# EMNG 3018 – Capstone Design

## Automated Indoor Navigation System for a Raspberry Pi-Based Robot

**Team Members:**
George Dobric [Team Lead]
Rosario Giardina
Nhat Tan Nguyen
Eddy Cheng
Wilton Bach

**Client:**
QRS
Represented by Frank Naccarato

**Supervisor:** Ryan Billinger

**Executive Summary**

At the request of QRS, the team began to create a robotic indoor navigation system which can perform certain basic tasks. Such an application would be practical in workplaces where simple tasks that require human intervention can be automated. In light of the COVID-19 pandemic, such a system could be in rising demand since it could serve as a tool to help companies reduce their risks of virus transmissions.

The team considered basing the system around an ESP32 microcontroller and have the HMI interface be based on Node-RED dashboard. However, the team figured that it would be more effective to base the system around a Raspberry Pi since it would consist of a direct data transfer system, as opposed to transferring the data from an ESP32 to a Raspberry Pi using a Pub-sub client such as MQTT. Using a Raspberry Pi, an ESP32 is used as a modem and is connected to the Raspberry Pi's USB port.

The navigation system consists of a remote-control car operated by a Raspberry Pi (which has an ESP32 connected to its USB port to serve as a modem) as well as four Bluetooth beacons (which will be ESP32s during the development phase). The team will add directional antennas to the ESP32 Bluetooth beacons once enough progress is made. At this point, an omnidirectional antenna will be added to the ESP32 connected to the Raspberry Pi on the remote-control car.

The RSSI signal strengths collected from each Bluetooth beacon to the Raspberry Pi must be converted to distance. To do so, the group will perform a series of measurements and associate measured distances with RSSI signal strength values. This data will be placed in an Excel document and a formula will be used in order to establish a method of converting RSSI values to distance values. Furthermore, the indoor navigation system will operate on a calibration system. The user will place the remote-control car in destination areas, or zones, and select a calibration prompt on the HMI. For instance, if the user desires four destination zones, the user should place the remote-control car in the four desired positions and press the calibrate prompt each time. This will store four RSSI values, from each Bluetooth beacon to the car, and will allow for proper navigation.

# Table of Contents

# 1.0 Problem Definition

Based on a written request which was received from the client, QRS, the primary objective is summarized in the following sections. The client's needs are taken into account alongside the team's principle focus which is to create a cost-effective solution which avoids unnecessary complexities.

## 1.1 Problem Statement

### Background

The final-semester capstone projects which are undertaken by students in an Electromechanical Engineering program at George Brown College are generally based around the field of automation. The automated indoor-navigation task which the team has taken up is further explained in the following 'Problem' subsection.

### Problem

The team is looking to implement a remote-control vehicle system, with automated indoor navigation, consisting of a remote-control car operated by a Raspberry Pi and five ESP32s which serve as Bluetooth beacons. The primary focus remains to be the programming of such an indoor navigation system. Using the four Bluetooth beacons that outline an area, the vehicle will be designed to allow for specific zone calibration in such an area. Each zone is to be calibrated by saving the signal strength of all four Bluetooth beacons. The zones will then be navigated by a programmed indoor navigation system which relies on the signal strengths of the four Bluetooth beacons. The ESP32 connected to the Raspberry Pi must have an adjusted BLE scan program uploaded to it in order to serve its purpose properly.

The team has assembled the vehicle which consists of a DC motor for the forward and reverse movement, three servo motors for steering left and right, and the movement of the robot head. All the components were purchased from "Adeept" as a kit. The vehicle is made of acrylic plates and includes four wheels, RGB LEDs, a Raspberry Pi 4 along side the motor driver board, an Adeept Motor HAT (which controls the motors) as well as an 18650-battery pack to power the car. The sensors used include an ultrasonic sensor for detecting obstacles, and a Pi camera for tracking the car's position. The team is looking to implement a portable battery pack system for the vehicle Raspberry Pi and the four ESP32 Bluetooth beacons. The remote-control car was programmed with the library given by the manufacturer. The vehicle is to be operated on a touchscreen, portable interface which displays a Node-RED dashboard. The dashboard will contain buttons in order to calibrate zones, select zones, and display the vehicle's position.

For the HMI, the team opted to use Node-RED dashboard on a portable, touchscreen device. Moreover, the team managed to use the ESP32's low-power Bluetooth modules as both a scanner and a beacon. Additionally, in order to detect and estimate the distance of the robot from each beacon, the team has implemented a Node-RED flow that will use an ESP32 as a modem.

The flow used was built upon an example flow which scans for advertising Bluetooth devices and lists each device's information as a raw, unformatted string. Using the flow, the Raspberry Pi will intermittently scan for advertising Bluetooth devices and provide a raw, unformatted string with information regarding each device's address, service UUID, and RSSI signal strength. The main functions of the flow are start/stop Bluetooth device scanning, device detection, and RSSI strength isolation which was done by creating a function which searches through each string. The signal strength decreases as the device gets further away from the user, so its distance is inferable by recording the RSSI values at set measured distances. A visual representation of the signal strength is included on the dashboard as a line chart.

## Scope

The project will include the programming of all ESP32 Bluetooth beacons as well as the Node-RED flow programming for the Raspberry Pi. Moreover, the development of an automated navigation system will be included as a theoretical system alongside the programmed code for its implementation. The team plans to finish the automated navigation system before March 12$^{th}$, 2021. However, due to missing components and the initial delay in shipment, this projection may be affected. The team aims to complete the project in its entirety by April 9$^{th}$, 2021.

## 1.2 Identification of Stakeholders

A list of individuals and groups who have a stake in the implementation and success of the indoor navigation system for a Raspberry Pi-based robot is outlined below in order of degree of impact.

*George Brown College Team*

The team consists of five students in their final semester of the Electromechanical Engineering Technology – Building Automation Program. The team is supervised by professor Ryan Billinger where six hours of lab time are dedicated per week to work on the project. The team's stake in the project is to put all of the members' skills to the test in order to have a representation of each individual's technical abilities.

*George Brown College*

The school uses the final-semester student capstone projects as an opportunity to represent the skills and abilities of George Brown students. Moreover, these projects can help with student employability which in turn is beneficial to the college.

*QRS*

QRS has provided the college with a list of tasks it seeks to see performed by students in order to analyse the way in which these tasks can be completed. With cost-effectiveness and efficiency in mind, there is a possibility for George Brown students to generate innovative ideas to tackle certain tasks at hand. Such solutions could be beneficial for QRS.

## 1.3 Design for the Automated Indoor Navigation System

The design of the remote-control car must take into account several design considerations as outlined below.

*Manufacturability*

The components which make up the remote-control car consist of acrylic plates, machinery parts, transmission parts, electronic parts, tools, two 18650 batteries and a Raspberry Pi. In terms of accessibility, every component is widely available and should not pose issues to the milestone deadlines. The acrylic plates, however, can be tailored in any matter which satisfies the client's needs. Such plates can be 3D printed for the sake of customization and control over the manufacturing time and capacity.

*Assembly*

The assembly of the remote-control car must be efficient and consistent in order to allow for the construction of the vehicle to be done in a timely manner. This can be achieved by ensuring there is no tedious, limiting factor during the assembly. Generally, the consumer-level assembly of such vehicles is subjected to the use of small, inefficient tools. In order to satisfy the client, the assembly should be done with more effective versions of the same tools. For instance, the hex wrench can be swapped with one which has a P-Handle, or a ratchet wrench can be used. The screwdrivers can be swapped for ratchet screwdrivers. Moreover, the socket wrench can be swapped for a ratchet socket wrench.

*Quality and Reliability*

The remote-control car is quite fragile and its development is simply a proof-of-concept for the task of automated indoor navigation. In regards to the remote-control car used in the development of this proof-of-concept, it is reliable however its quality can be improved upon should the client be satisfied with the development. The commercial application of such a system should seek to implement a remote-control car which is larger in all dimensions to allow for faster navigation and more capable task performance. Furthermore, a larger vehicle will allow for a more rigid structure suited for commercial environments. A larger vehicle would require larger plates, machinery and transmission parts. Additionally, more batteries would be necessary to support such a system.

*Maintenance and Serviceability*

The remote-control car system is entirely modular and maintenance can be done with great ease. Should a plate, machinery, transmission, or electrical part need replacement – the user can simply replace the component in question. In order to make maintenance of the vehicle efficient and suitable in a commercial environment, the tools used should be upgraded to ratchet screwdrivers, socket wrenches and wrenches for quick assembly and disassembly.

*Safety*

The development of the small-scale proof-of-concept vehicle does not pose any significant safety issues for its operator or the people residing in its zone of operation. Should the client request the construction of such a system with a large-scale vehicle, some safety issues may be introduced. The most notable potential issue with a large-scale vehicle is a high-speed collision between the vehicle and an individual or item in the zone of operation. This issue can be mitigated by two methods which can be implemented. One method is to limit the speed of the vehicle to a safe speed relative to the size of the vehicle. This would ensure that should a possible collision commence, the individual in the zone should have more time to react and get out of the way. Moreover, should a collision occur, the force of impact will be much less at a low speed. Another method of avoiding collisions is to add redundant individual/object detection systems on the vehicle. This can be done with a combination of ultrasonic sensors and a camera.

*Usability*

The remote-control car will be operated by one user through a tablet which must be connected to the vehicle's Raspberry Pi through a VNC client. The tablet will displaying Node-RED dashboard which will contain buttons for zone calibration and zone selection.

## 1.4 Functions

Functions indicate the purpose of the design, and therefore are requirements that must be met. This section contains basic functions of the vehicle transmission and sensors.

*Primary*

- The Raspberry Pi will power the motor to propel the vehicle forward should the Node-RED logic signal it to do so
- The transmission will deliver power from the motor to the vehicle's wheels
- The servo motor will allow for the rotation of the front wheels

*Secondary*

- The powering of the vehicle will be done by a power bank

*Unintended*

- The Raspberry Pi's operation will generate heat

## 1.5 Objectives

This section contains the project's categorized objectives obtained through the client's written request and the team's brainstorming.

*Physical*

- The Raspberry Pi vehicle must be compact
- The Bluetooth beacons must be mounted on moveable stands
- Both the ESP32s and the Rasbperry Pi must be battery-powered

*Financial*

- The design should be cost effective, provided the RSSI signal strength detection is accurate at large ranges

Performance

- The design should be intuitive for the user operating the tablet display
- The design should allow for relatively quick maneuvering of the vehicle
- The design should allow for complete autonomous indoor navigation

## 1.6 Constraints

The following are a list of restrictions that the design must satisfy.

*Design Rules*

- Once the user selects their desired destination zone, the vehicle must autonomously navigate to its destination
- The navigation must be based on a system of Bluetooth RSSI signal strength detection
- The Raspberry Pi and the four ESP32 Bluetooth beacons must be battery powered
- The Bluetooth beacons must be mounted on maneuverable stands
- The system must allow for user zone calibration
- The vehicle must perform a 180 degree turn once it has arrived at its destination

*Client Specified*

- The vehicle must be able to perform tasks
- The system must be adaptable to different indoor environments

## 1.7 Service Environment

This section contains the conditions in which the Raspberry Pi vehicle will operate. The final design will be operated in standard indoor environments where air quality and conditions are to be consistent. The physical, human, and virtual environments are the factors taken into consideration for this section.

*Physical Environment*

The Raspberry Pi will be situated on a standard non-enclosed vehicle and is designed to operate in indoor environments which uphold standard expected indoor air quality and conditions. Temperature, humidity and vehicle structure are the factors taken into account for the physical environment.

- *Temperature*: the average room temperature must be in the range of 20°C and 25°C
- *Humidity*: Relative humidity in the average room must be between 30% and 50%
- *Vehicle Structure*: The Raspberry Pi will be mounted on a non-enclosed vehicle. This proof-of-concept design does not expect any forces to be exerted on the vehicle.

*Human Environment*

The Raspberry Pi vehicle will be operated wirelessly by a user operating a tablet interface. The human environment factor considered in this section is experience.

- *Experience*: The user must be able to understand how to troubleshoot the system should there be a minor connectivity error. This would primarily concern the tablet's VNC viewer connection with the Raspberry Pi.

*Virtual Environment*

The Raspberry Pi relies on the ESP32 Bluetooth Low Energy scanner plugged into it in order to read the RSSI signal strength of the four ESP32 Bluetooth beacons. In order to minimize signal interference, it would be desirable to operate such a system in an environment which has relatively low amounts of Bluetooth devices and signal-blocking materials.

## 1.8 Human Factors

An intuitive and simple user interface to operate the automated indoor navigation system will be essential. For this section, knowledge of networking, Raspberry Pi, Arduino IDE, and Node-RED will be examined.

*Networking*

Although the automated indoor navigation system is to be designed for simple operation such that any user can operate it intuitively, knowledge of networking is fundamental if one wishes to have the ability to troubleshoot and alter the system to their desired outcomes. The networking concerned with the system is primarily in the focus of connectivity. The ESP32 Bluetooth Beacons must have connectivity with the ESP32 Bluetooth scanner plugged into the

Raspberry Pi. Moreover, the Raspberry Pi must have connectivity with the user tablet by means of VNC viewer.

*Raspberry Pi*

In case of the vehicle's malfunction, it would be ideal for the user to have knowledge of the Raspberry Pi board's inputs and outputs. It would be recommended that the user is familiar with Raspberry Pi's and their pinouts.

*Arduino IDE*

Although the system is designed to operate intuitively, should the user wish to make alterations to the advertising ESP32s or the scanning ESP32 — a knowledge of Arduino IDE would be suitable for such a task.

*Node-RED*

Since the entire automated indoor navigation system runs on a Node-RED program, it would be optimal for the user to have experience with Node-RED should they wish to make changes to the system. The Node-RED flow gathers local RSSI signals, isolates the desired signals, and automates the indoor navigation.

# 2.0 Detailed Design

The following sections provide information regarding the final design of the automated indoor navigation system. It was decided to operate the system based on Bluetooth Low Energy RSSI signals which are advertised from four ESP32 Bluetooth beacons. Such a system allows for adaptability and scalability with possible operation in nearly any indoor environment.

## 2.1 Signal to Distance Conversion

Once the RSSI signal of an advertising Bluetooth device was isolated, a table was created and twenty RSSI signal strengths were recorded for varying distances. Averaging the signal strength for each distance, one observation which was made is that the greater the distance between the ESP32 and the advertising Bluetooth device, the less of a difference there is in signal strength. Another observation which was made is the difference between the lowest and highest signal at each distance was not consistent. At 0, 0.5, 1, 2, and 3 meters the fluctuation differences were 3, 5, 8, 5, and 7 decibels respectively. However, at 1.5, 2.5, and 5.5 meters the differences were 14, 15, and 13 decibels respectively. These measurements were done using a Bluetooth speaker, which sends its signal in an omnidirectional manner. For this reason, these inconsistencies can be expected and should be minimized using directional Bluetooth advertising.

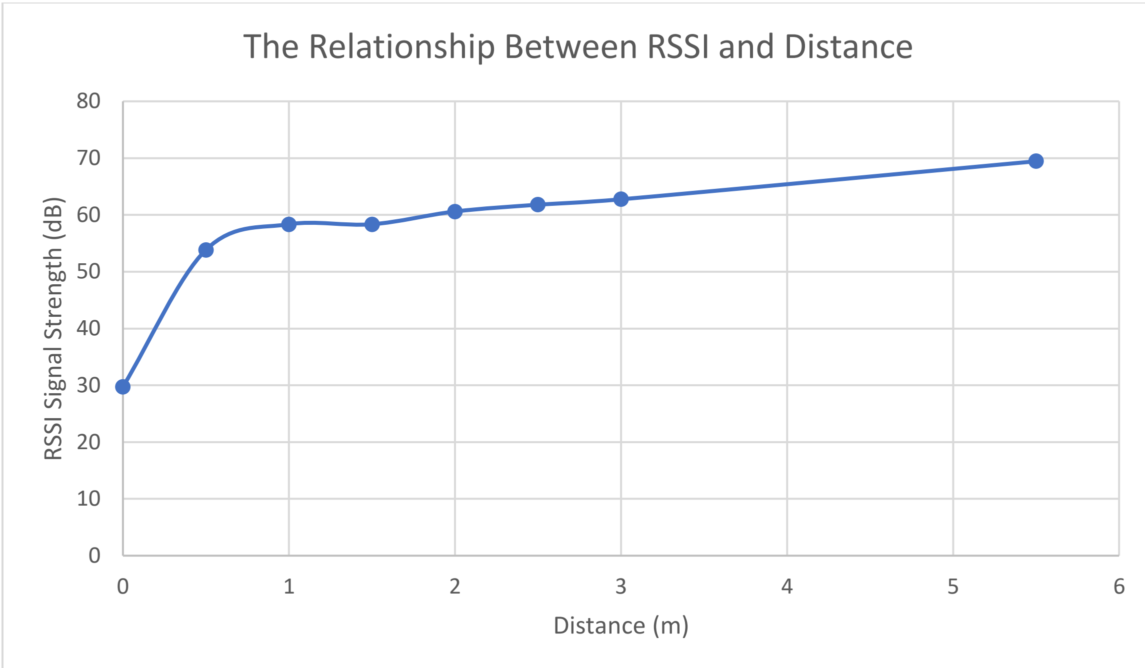



*Figure 1 Graph plotting the average signal strength recorded at varying distances*

As can be seen in the figure above, there is an exponential decrease in the signal strength differences between distances the further the Bluetooth device is from the ESP32. This can create issues when one is attempting to create an indoor navigation system which solely relies on signal strength values. This issue can be resolved by implementing directional Bluetooth antennas and incorporating the vehicle's speed in order to account for distance coverage. The directional antennas provide a more consistent signal by providing a coned zone of connection, relative to the omnidirectional Bluetooth radial zone which can be more inaccurate for the purposes of this project. The incorporation of the vehicle's speed by using the motor rotation as well as measured speed values can help to serve as a redundant source of positioning information which can supplement the signal strength navigation.


*Figure 2 RSSI Signal Strength Mapping with Advertising Bluetooth Speaker*

| Distance (m) | Signal Strength Avg (dB) | Signal Strength Min, Max (c | 0.5 m | 1 m | 1.5 m | 2 m | 2.5 m | 3 m | | 0 m | | | 5.5 m |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0.5 | 47.15 | 45, 50 (Delta = 5) | 46 | 50 | 57 | 64 | 66 | 63 | | 30 | | | 70 |
| 1 | 53.85 | 50, 58 (Delta = 8) | 47 | 52 | 59 | 60 | 57 | 62 | | 30 | | | 70 |
| 1.5 | 58.35 | 54, 68 (Delta = 14) | 46 | 50 | 58 | 63 | 71 | 64 | | 29 | | | 78 |
| 2 | 60.6 | 59, 64 (Delta = 5) | 49 | 50 | 59 | 61 | 60 | 67 | | 29 | | | 67 |
| 2.5 | 61.8 | 56, 71 (Delta = 15) | 46 | 57 | 68 | 61 | 57 | 61 | | 29 | | | 71 |
| 3 | 62.75 | 60, 67 (Delta = 7) | 46 | 55 | 55 | 63 | 60 | 60 | | 30 | | | 71 |
| | | | 48 | 57 | 55 | 61 | 57 | 63 | | 30 | | | 69 |
| | | | 48 | 58 | 55 | 59 | 56 | 60 | | 30 | | | 67 |
| | | | 50 | 57 | 67 | 61 | 61 | 64 | | 30 | | | 67 |
| | | | 50 | 58 | 55 | 59 | 68 | 61 | | 30 | | | 78 |
| 5.5 | 69.45 | 65, 78 (Delta = 13) | 45 | 51 | 66 | 60 | 56 | 60 | | 30 | | | 71 |
| 0 | 29.75 | 28, 31 (Delta = 3) | 49 | 58 | 65 | 61 | 69 | 63 | | 29 | | | 72 |
| | | | 49 | 51 | 58 | 60 | 69 | 62 | | 31 | | | 65 |
| | | | 45 | 53 | 55 | 59 | 57 | 64 | | 30 | | | 67 |
| | | | 47 | 51 | 58 | 61 | 69 | 63 | | 30 | | | 66 |
| | | | 47 | 53 | 55 | 59 | 61 | 63 | | 30 | | | 72 |
| | | | 46 | 57 | 55 | 60 | 61 | 64 | | 28 | | | 66 |
| | | | 46 | 51 | 58 | 60 | 57 | 63 | | 29 | | | 67 |
| | | | 47 | 53 | 55 | 61 | 61 | 63 | | 30 | | | 66 |
| | | | 46 | 55 | 54 | 59 | 63 | 65 | | 31 | | | 69 |
| | | **Average** | 47.15 | 53.85 | 58.35 | 60.6 | 61.8 | 62.75 | | 29.75 | | | 69.45 |
| | | **Minimum** | 45 | 50 | 54 | 59 | 56 | 60 | | 28 | | | 65 |
| | | **Maxmimum** | 50 | 58 | 68 | 64 | 71 | 67 | | 31 | | | 78 |

*Figure 3 RSSI Signal Strength Mapping from Bluetooth Speaker Measurements and Raspberry Pi Signal Recordings*

As can be seen in *Figure 2*, an advertising Bluetooth Speaker was used in accordance with an ESP32 (uploaded with Bluetooth Low Energy scan code) plugged into a Raspberry Pi in order to map the RSSI signal strength. The intention of this mapping was to create a formula from which distance can be inferred once a signal strength is obtained. Twenty signal strengths were recorded at each distance in order to obtain an average as well as peak and trough values.

## 2.2 Indoor Navigation Design

The automated indoor navigation was designed to operate within an area with four destination zones, six tables, and an isle of longitudinal transportation. The design is represented in the following figure:

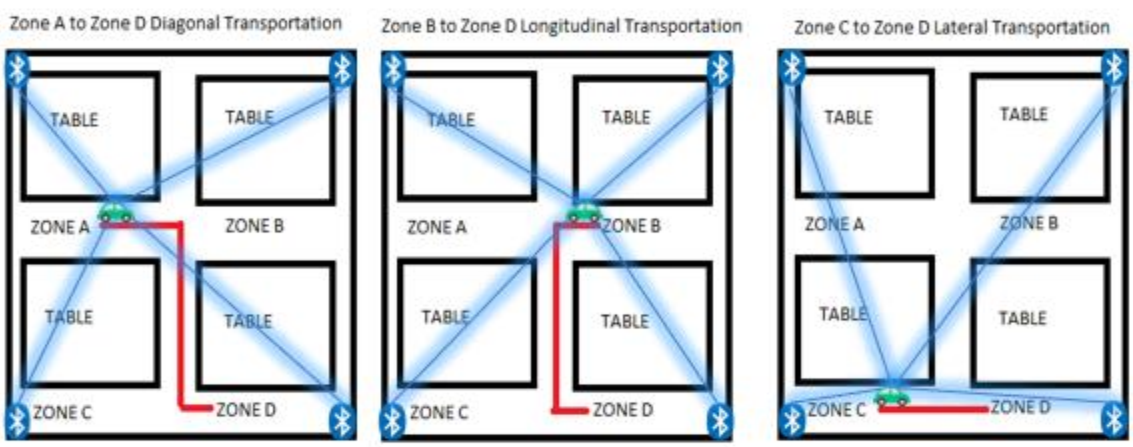



*Figure 4 Diagram displaying the three types of movement in order to travel to Zone D: Lateral, Longitudinal, and Diagonal*

The following code was developed to simulate automated indoor navigation. The purpose of this development was to prepare the navigation code so that once the RSSI signal collection from four of the ESP32 Bluetooth beacons was complete and ready, the signal collection could be easily fitted into this code which would result in time saved for the team. The code has four blocks, each of which is dedicated to each zone, allowing for longitudinal, lateral, and diagonal transportation. The simulation code uses a mapping of a ten-by-ten system, with zone A being referred to by the coordinates (2,2), zone B by (8, 2), zone C by (2, 8), and zone D by (8,8). The variables assigned form lines 6 to 13 of the code were not used, however this is where the RSSI signal collection would be assigned. The collected signals could be mapped out to determine x and y values for an obtained signal, this would require several measurements to be done and recorded in order to determine the x and y correlation with a single signal.

```
/*  A        B



    C        D*/
msg.aX = 0;
msg.aY = 0;
msg.bX = 10;
msg.bY = 0;
msg.cX = 0;
msg.cY = 10;
msg.dX = 10;
msg.dY = 10;
if (msg.startAtoB == null)
    msg.startAtoB == true;
if (msg.startAtoC == null)
    msg.startAtoC == true;
if (msg.startAtoD == null)
    msg.startAtoD == true;
if (msg.startBtoA == null)
    msg.startBtoA == true;
if (msg.startBtoC == null)
    msg.startBtoC == true;
if (msg.startBtoD == null)
    msg.startBtoD == true;
if (msg.startCtoA == null)
    msg.startCtoA == true;
if (msg.startCtoB == null)
    msg.startCtoB == true;
if (msg.startCtoD == null)
    msg.startCtoD == true;
if (msg.startDtoA == null)
    msg.startDtoA == true;
if (msg.startDtoB == null)
    msg.startDtoB == true;
if (msg.startDtoC == null)
    msg.startDtoC == true;

if (msg.raspX == null && msg.raspY == null){
    msg.raspX == 2;
    msg.raspY == 2;
}

// Destination Zone D
if (msg.zoneD == true){

    // Diagonal transportation section begins
    // zone A to D transportation begins
    if (msg.destA == true){
        if (msg.startAtoD == true)
            msg.AtoD = 1;
        do {
```

```
            msg.raspX += 1; //this is where motor will be actuated

        } while (msg.raspX < 5 && msg.AtoD == 1)
        if (msg.raspX == 5 && msg.raspY == 2)
            msg.AtoD = 2;
        do {
            //this is where there will be  one-time car rotation
            msg.raspY += 1; //this is where the motor will be actuted
        } while (msg.raspY < 8 && msg.AtoD == 2)
        if (msg.raspY == 8 && msg.raspX == 5)
            msg.AtoD = 3;
        do {
            //this is where there will be  one-time car rotation
            msg.raspX += 1; //this is where motor will be actuated

        } while (msg.raspX < 8 && msg.AtoD == 3)
        if (msg.raspX == 8 && msg.raspY == 8)
            msg.AtoD = 4;
        if (msg.AtoD == 4){
            //initiate 180 degree stationary rotation
        }
    }
    // zone A to D transportation ends
    // Diagonal transportation section ends


    // Longitudinal transportation section begins
    // Zone B to D transportation begins
    if (msg.destB == true){
        if (msg.startBtoD == true)
            msg.BtoD = 1;
        do {
            msg.raspX -= 1; //this is where motor will be actuated

        } while (msg.raspX > 5 && msg.BtoD == 1)
        if (msg.raspX == 5 && msg.raspY == 2)
            msg.BtoD = 2;
        do {
            //this is where there will be  one-time car rotation
            msg.raspY += 1; //this is where the motor will be actuted
        } while (msg.raspY < 8 && msg.BtoD == 2)
        if (msg.raspY == 8 && msg.raspX == 5)
            msg.BtoD = 3;
        do {
            //this is where there will be  one-time car rotation
            msg.raspX += 1; //this is where motor will be actuated

        } while (msg.raspX < 8 && msg.BtoD == 3)
        if (msg.raspX == 8 && msg.raspY == 8)
            msg.BtoD = 4;
        if (msg.BtoD == 4){
            //initiate 180 degree stationary rotation
```

```
        }
    }
    // zone B to D transportation ends
    // Longitudinal transportation section ends

    // Lateral transportation section begins
    // Zone C to D transportation begins
    if (msg.destC == true){
        if (msg.startCtoD == true)
            msg.CtoD = 1;
        do {
            msg.raspX += 1; //this is where motor will be actuated

        } while (msg.raspX < 8 && msg.CtoD == 1)
        if (msg.raspX == 8 && msg.raspY == 8)
            msg.CtoD = 2;
        if (msg.CtoD == 2){
            //initiate 180 degree stationary rotation
        }
    }
    // zone C to D transportation ends
    // Lateral transportation section ends

}

// Destination Zone C
if (msg.zoneC == true){

    // Diagonal transportation section begins
    // zone B to C transportation begins
    if (msg.destB == true){
        if (msg.startBtoC == true)
            msg.BtoC = 1;
        do {
            msg.raspX -= 1; //this is where motor will be actuated

        } while (msg.raspX > 5 && msg.BtoC == 1)
        if (msg.raspX == 5 && msg.raspY == 2)
            msg.BtoC = 2;
        do {
            //this is where there will be  one-time car rotation
            msg.raspY += 1; //this is where the motor will be actuted
        } while (msg.raspY < 8 && msg.BtoC == 2)
        if (msg.raspY == 8 && msg.raspX == 5)
            msg.BtoC = 3;
        do {
            //this is where there will be  one-time car rotation
            msg.raspX -= 1; //this is where motor will be actuated

        } while (msg.raspX > 2 && msg.BtoC == 3)
        if (msg.raspX == 2 && msg.raspY == 8)
            msg.BtoC = 4;
```

```
        if (msg.BtoC == 4){
            //initiate 180 degree stationary rotation
        }
    }
    // zone B to C transportation ends
    // Diagonal transportation section ends


    // Longitudinal transportation section begins
    // Zone A to C transportation begins
    if (msg.destA == true){
        if (msg.startAtoC == true)
            msg.AtoC = 1;
        do {
            msg.raspX += 1; //this is where motor will be actuated

        } while (msg.raspX < 5 && msg.AtoC == 1)
        if (msg.raspX == 5 && msg.raspY == 2)
            msg.AtoC = 2;
        do {
            //this is where there will be  one-time car rotation
            msg.raspY += 1; //this is where the motor will be actuted
        } while (msg.raspY < 8 && msg.AtoC == 2)
        if (msg.raspY == 8 && msg.raspX == 5)
            msg.AtoC = 3;
        do {
            //this is where there will be  one-time car rotation
            msg.raspX -= 1; //this is where motor will be actuated

        } while (msg.raspX > 2 && msg.AtoC == 3)
        if (msg.raspX == 2 && msg.raspY == 8)
            msg.AtoC = 4;
        if (msg.AtoC == 4){
            //initiate 180 degree stationary rotation
        }
    }
    // zone A to C transportation ends
    // Longitudinal transportation section ends

    // Lateral transportation section begins
    // Zone D to C transportation begins
    if (msg.destD == true){
        if (msg.startDtoC == true)
            msg.DtoC = 1;
        do {
            msg.raspX -= 1; //this is where motor will be actuated

        } while (msg.raspX > 2 && msg.DtoC == 1)
        if (msg.raspX == 2 && msg.raspY == 8)
            msg.DtoC = 2;
        if (msg.DtoC == 2){
            //initiate 180 degree stationary rotation
```

```
        }
    }
    // zone D to C transportation ends
    // Lateral transportation section ends

}

// Destination Zone B
if (msg.zoneB == true){

    // Diagonal transportation section begins
    // zone C to B transportation begins
    if (msg.destC == true){
        if (msg.startCtoB == true)
            msg.CtoB = 1;
        do {
            msg.raspX += 1; //this is where motor will be actuated

        } while (msg.raspX < 5 && msg.CtoB == 1)
        if (msg.raspX == 5 && msg.raspY == 8)
            msg.CtoB = 2;
        do {
            //this is where there will be  one-time car rotation
            msg.raspY -= 1; //this is where the motor will be actuted
        } while (msg.raspY > 2 && msg.CtoB == 2)
        if (msg.raspY == 2 && msg.raspX == 5)
            msg.CtoB = 3;
        do {
            //this is where there will be  one-time car rotation
            msg.raspX += 1; //this is where motor will be actuated

        } while (msg.raspX < 8 && msg.CtoB == 3)
        if (msg.raspX == 8 && msg.raspY == 2)
            msg.CtoB = 4;
        if (msg.CtoB == 4){
            //initiate 180 degree stationary rotation
        }
    }
    // zone C to B transportation ends
    // Diagonal transportation section ends


    // Longitudinal transportation section begins
    // Zone D to B transportation begins
    if (msg.destD == true){
        if (msg.startDtoB == true)
            msg.DtoB = 1;
        do {
            msg.raspX -= 1; //this is where motor will be actuated

        } while (msg.raspX > 5 && msg.DtoB == 1)
        if (msg.raspX == 5 && msg.raspY == 8)
```

```
            msg.DtoB = 2;
        do {
            //this is where there will be  one-time car rotation
            msg.raspY -= 1; //this is where the motor will be actuted
        } while (msg.raspY > 2 && msg.DtoB == 2)
        if (msg.raspY == 2 && msg.raspX == 5)
            msg.DtoB = 3;
        do {
            //this is where there will be  one-time car rotation
            msg.raspX += 1; //this is where motor will be actuated

        } while (msg.raspX < 8 && msg.DtoB == 3)
        if (msg.raspX == 8 && msg.raspY == 2)
            msg.DtoB = 4;
        if (msg.DtoB == 4){
            //initiate 180 degree stationary rotation
        }
    }
    // zone D to B transportation ends
    // Longitudinal transportation section ends

    // Lateral transportation section begins
    // Zone A to B transportation begins
    if (msg.destA == true){
        if (msg.startAtoB == true)
            msg.AtoB = 1;
        do {
            msg.raspX += 1; //this is where motor will be actuated

        } while (msg.raspX < 8 && msg.AtoB == 1)
        if (msg.raspX == 8 && msg.raspY == 2)
            msg.AtoB = 2;
        if (msg.AtoB == 2){
            //initiate 180 degree stationary rotation
        }
    }
    // zone A to B transportation ends
    // Lateral transportation section ends

}

// Destintion Zone A
if (msg.zoneA == true){

    // Diagonal transportation section begins
    // zone D to A transportation begins
    if (msg.destD == true){
        if (msg.startDtoA == true)
            msg.DtoA = 1;
        do {
            msg.raspX -= 1; //this is where motor will be actuated

```

```
        } while (msg.raspX > 5 && msg.DtoA == 1)
        if (msg.raspX == 5 && msg.raspY == 8)
            msg.DtoA = 2;
        do {
            //this is where there will be  one-time car rotation
            msg.raspY -= 1; //this is where the motor will be actuted
        } while (msg.raspY > 2 && msg.DtoA == 2)
        if (msg.raspY == 2 && msg.raspX == 5)
            msg.DtoA = 3;
        do {
            //this is where there will be  one-time car rotation
            msg.raspX -= 1; //this is where motor will be actuated

        } while (msg.raspX > 2 && msg.DtoA == 3)
        if (msg.raspX == 2 && msg.raspY == 2)
            msg.DtoA = 4;
        if (msg.DtoA == 4){
            //initiate 180 degree stationary rotation
        }
    }
    // zone D to A transportation ends
    // Diagonal transportation section ends


    // Longitudinal transportation section begins
    // Zone C to A transportation begins
    if (msg.destC == true){
        if (msg.startCtoA == true)
            msg.CtoA = 1;
        do {
            msg.raspX += 1; //this is where motor will be actuated

        } while (msg.raspX < 5 && msg.CtoA == 1)
        if (msg.raspX == 5 && msg.raspY == 8)
            msg.CtoA = 2;
        do {
            //this is where there will be  one-time car rotation
            msg.raspY -= 1; //this is where the motor will be actuted
        } while (msg.raspY > 2 && msg.CtoA == 2)
        if (msg.raspY == 2 && msg.raspX == 5)
            msg.CtoA = 3;
        do {
            //this is where there will be  one-time car rotation
            msg.raspX -= 1; //this is where motor will be actuated

        } while (msg.raspX > 2 && msg.CtoA == 3)
        if (msg.raspX == 2 && msg.raspY == 2)
            msg.CtoA = 4;
        if (msg.CtoA == 4){
            //initiate 180 degree stationary rotation
        }
    }
```

```
    // zone C to A transportation ends
    // Longitudinal transportation section ends

    // Lateral transportation section begins
    // Zone B to A transportation begins
    if (msg.destB == true){
        if (msg.startBtoA == true)
            msg.BtoA = 1;
        do {
            msg.raspX -= 1; //this is where motor will be actuated

        } while (msg.raspX > 2 && msg.BtoA == 1)
        if (msg.raspX == 2 && msg.raspY == 2)
            msg.BtoA = 2;
        if (msg.BtoA == 2){
            //initiate 180 degree stationary rotation
        }
    }
    // zone B to A transportation ends
    // Lateral transportation section ends

}
return msg;
```

The following bit of code was used in a Node-RED function node in order to print the obtained RSSI signals from each ESP32 Bluetooth beacon into the debug log.

```
if (msg.add2 == msg.add3 && msg.sig != null){
    msg.payload = msg.sig;
    node.warn(msg.sig);
}
if (msg.add2 == msg.add4 && msg.sig2 != null){
    msg.payload = msg.sig2;
    node.warn(msg.sig2);
}
if (msg.add2 == msg.add5 && msg.sig3 != null){
    msg.payload = msg.sig3;
    node.warn(msg.sig3);
}
if (msg.add2 == msg.add6 && msg.sig4 != null){
    msg.payload = msg.sig4;
    node.warn(msg.sig4);
}
return msg;
```

The following code was used in a Node-RED function node in order to isolate the RSSI signals of the four ESP32 Bluetooth beacons by searching through their raw strings and isolating the signal strength value. The advertised devices were isolated by simply searching for the first two characters of each ESP32's address. One could have typed in the whole addresses but this was unnecessary.

```
var sLength = msg.payload.length;

for (var i = 0; i<sLength; i++){

    if (msg.payload[i] == 'U' && msg.payload[i+1] == 'U' &&
msg.payload[i+2] == 'I' && msg.payload[i+3] == 'D'){// && msg.payload[i+4]
== 'e' && msg.payload[i+5] == 's' && msg.payload[i+6] == 's'){

        msg.add2 =msg.payload[i+6] + msg.payload[i+7] + msg.payload[i+8] +
msg.payload[i+9] + msg.payload[i+10] + msg.payload[i+11] +
msg.payload[i+12] + msg.payload[i+13];// + msg.payload[i+17] +
msg.payload[i+18] + msg.payload[i+19] + msg.payload[i+20] +
msg.payload[i+21] + msg.payload[i+22] + msg.payload[i+23] +
msg.payload[i+24] + msg.payload[i+25];
        ///for (var = k; k<msg.add2.length; k++){
            if (msg.add2[0] == '4' && msg.add2[1] == 'f'){// &&
msg.add2[2] == ":" && msg.add2[3] == '2' && msg.add2[4] == 'c' &&
msg.add2[5] == ":" && msg.add2[6] == 'b' && msg.add2[7] == '6' &&
msg.add2[8] == ":" && msg.add2[9] == '4' && msg.add2[10] == '9' &&
msg.add2[11] == ":" && msg.add2[12] == 'e' && msg.add2[13] == '1' &&
msg.add2[14] == ":" && msg.add2[15] == '4' && msg.add2[16] == 'f' ){
                msg.add3 = msg.add2;
                msg.test =
"aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa";
                msg.sig = msg.payload[111] + msg.payload[112];//116,117

            }
            if (msg.add2[0] == '5' && msg.add2[1] == 'f'){// &&
msg.add2[2] == ":" && msg.add2[3] == '2' && msg.add2[4] == 'c' &&
msg.add2[5] == ":" && msg.add2[6] == 'b' && msg.add2[7] == '6' &&
msg.add2[8] == ":" && msg.add2[9] == '4' && msg.add2[10] == '9' &&
msg.add2[11] == ":" && msg.add2[12] == 'e' && msg.add2[13] == '1' &&
msg.add2[14] == ":" && msg.add2[15] == '4' && msg.add2[16] == 'f' ){
                msg.add4 = msg.add2;
                msg.test =
"aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa";
                msg.sig2 = msg.payload[111] + msg.payload[112];//116,117

            }
            if (msg.add2[0] == '6' && msg.add2[1] == 'f'){// &&
msg.add2[2] == ":" && msg.add2[3] == '2' && msg.add2[4] == 'c' &&
msg.add2[5] == ":" && msg.add2[6] == 'b' && msg.add2[7] == '6' &&
msg.add2[8] == ":" && msg.add2[9] == '4' && msg.add2[10] == '9' &&
msg.add2[11] == ":" && msg.add2[12] == 'e' && msg.add2[13] == '1' &&
msg.add2[14] == ":" && msg.add2[15] == '4' && msg.add2[16] == 'f' ){
```

```
                msg.add5 = msg.add2;
                msg.test = 
"aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa";
                msg.sig3 = msg.payload[111] + msg.payload[112];//116,117

            }
            if (msg.add2[0] == '7' && msg.add2[1] == 'f'){// && 
msg.add2[2] == ":" && msg.add2[3] == '2' && msg.add2[4] == 'c' && 
msg.add2[5] == ":" && msg.add2[6] == 'b' && msg.add2[7] == '6' && 
msg.add2[8] == ":" && msg.add2[9] == '4' && msg.add2[10] == '9' && 
msg.add2[11] == ":" && msg.add2[12] == 'e' && msg.add2[13] == '1' && 
msg.add2[14] == ":" && msg.add2[15] == '4' && msg.add2[16] == 'f' ){
                msg.add6 = msg.add2;
                msg.test = 
"aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa";
                msg.sig4 = msg.payload[111] + msg.payload[112];//116,117

            }
        //}
        /*if (msg.add2 == "6e:91:44:e1:e1:a9"){//"08:2C:B6:49:E1:4F"){
            msg.add = msg.add2;
            msg.sig = "61";
            for (var j = 0; j<sLength; j++){
            if (msg.payload[j] == 'R' && msg.payload[j+1] == 's' && 
msg.payload[j+2] == 's' && msg.payload[j+3] == 'i'){
                msg.sig = "52";//msg.payload[j+7] + msg.payload[j+8];

                //msg.payload = 
"aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa";
            }
            }
            }*/
        }
    }



return msg;
```

The following change node was used in order to convert the string number of the signal strength into an actual number.

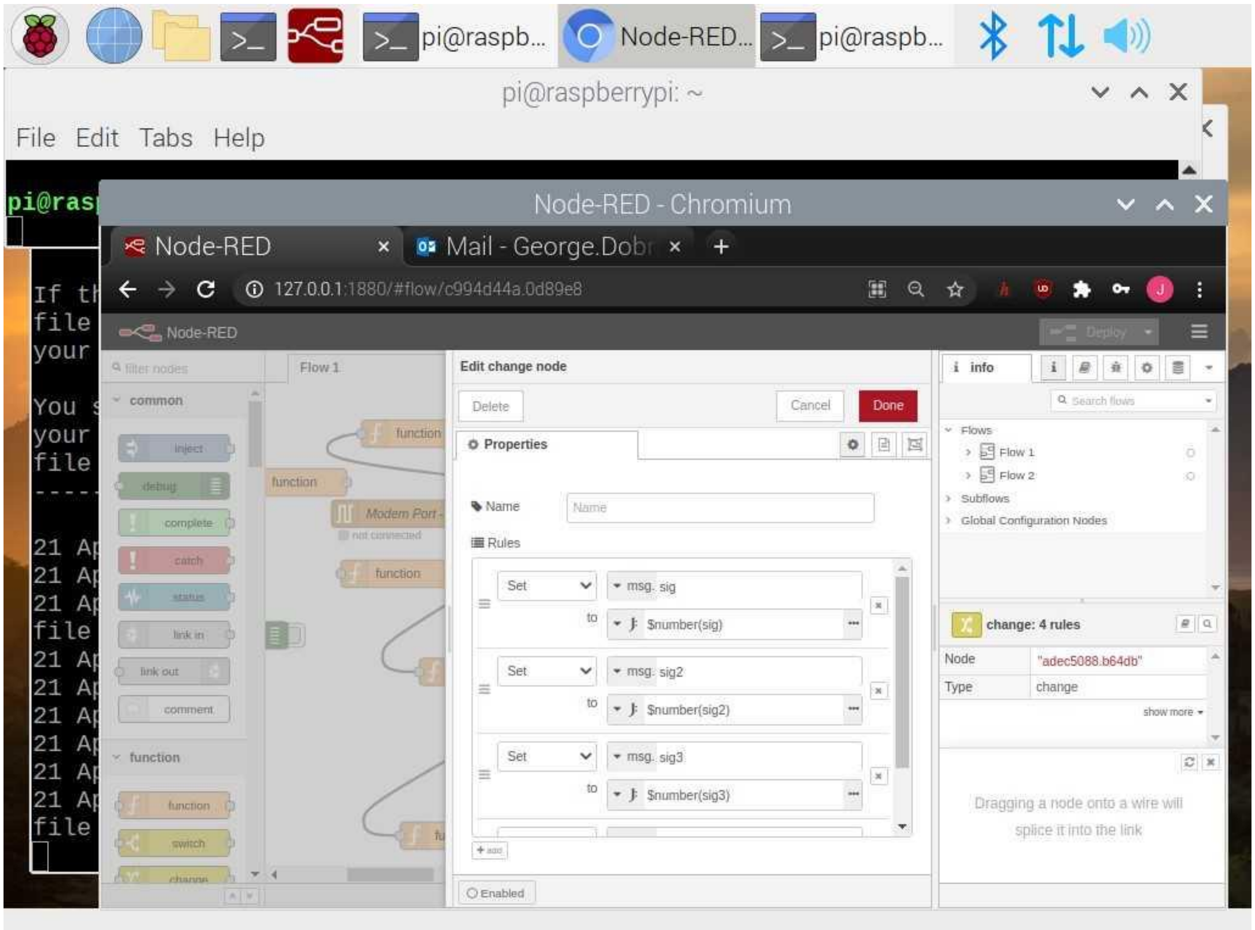



*Figure 5 Change node which converts string numbers into numbers*

The following is a serial node which identifies the Raspberry Pi USB port which the ESP32 scanner is plugged into.

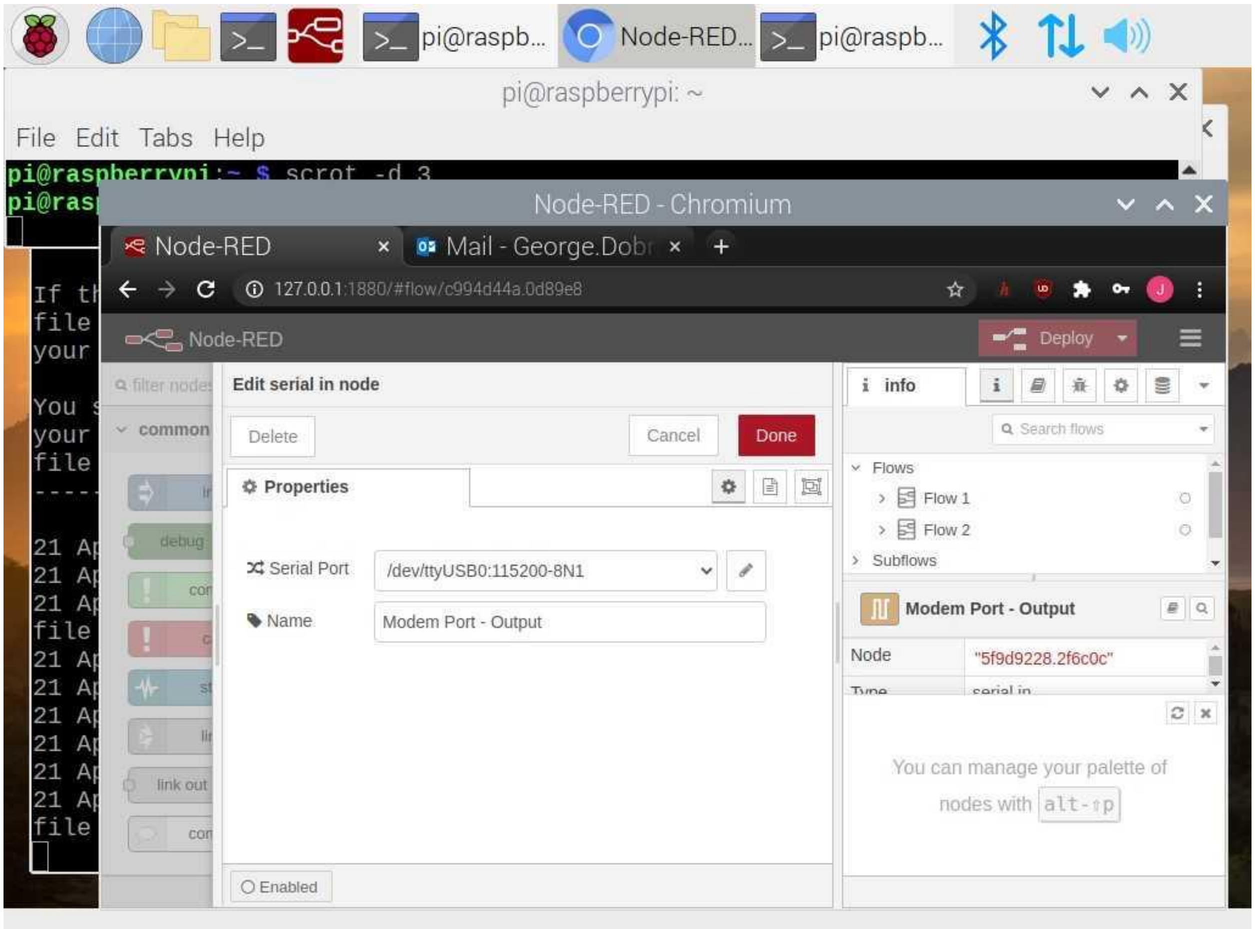



*Figure 6 Serial node which identifies the Raspberry Pi USB port that connects to the ESP32 Bluetooth scanner*

The following is a screenshot of the Node-RED flow for the automated indoor navigation system. There are some unconnected nodes which remained from system testing.

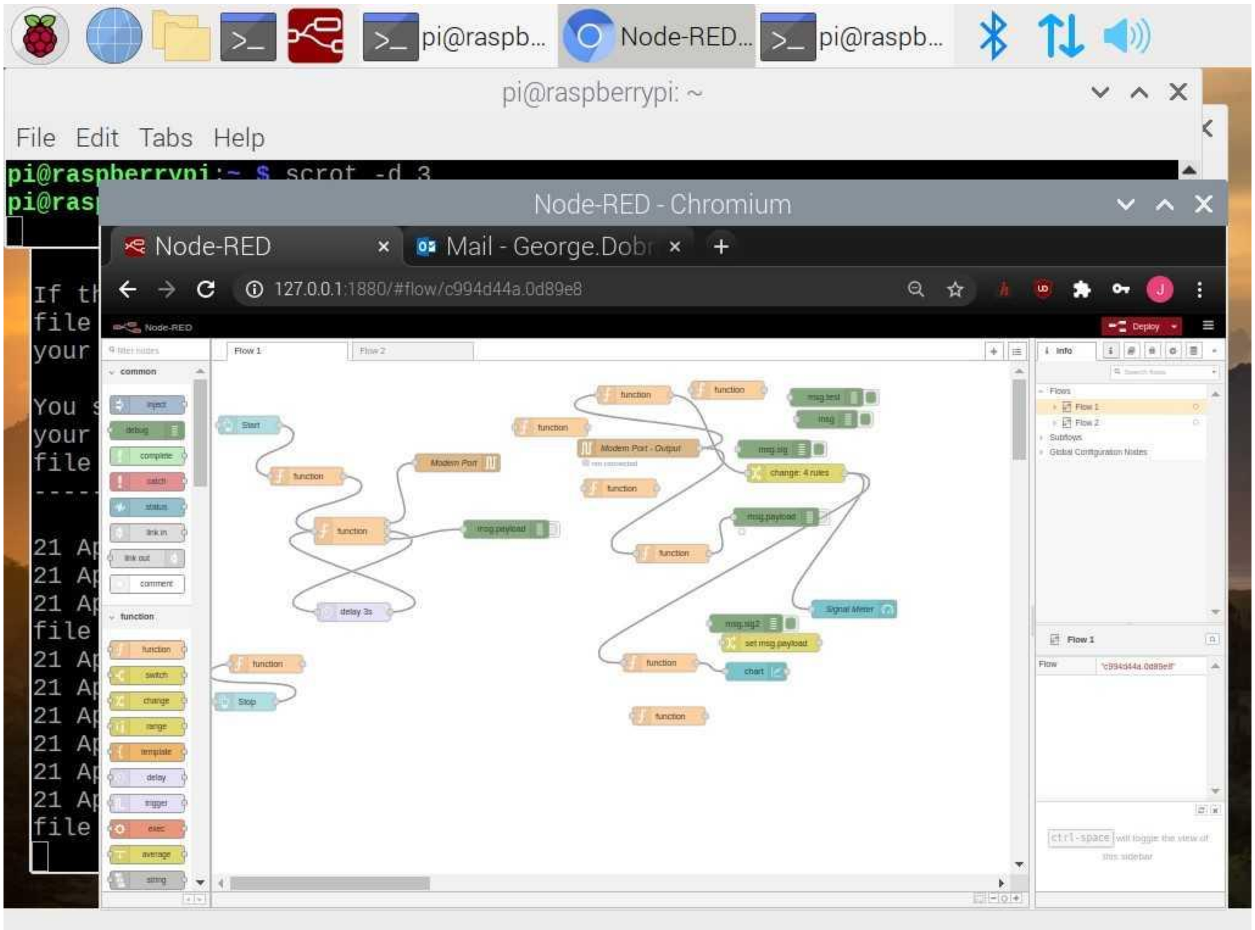



*Figure 7 Node-RED flow for the automated indoor navigation system*

## 2.3 Physical System Design

Before initiating the development of an automated indoor navigation system, it was important to brainstorm ideas and come to a certain project plan. The team discussed and theorised potential systems to fulfill the client's demands and concluded by deciding to navigate the system using Bluetooth Low Energy RSSI signal strength. Such a system allows for adaptability and scalability should the client wish to pursue this proof of concept.

## 2.4 Automated Indoor Navigation Simulation

The automated indoor navigation system was initiated by creating a simulation code for such a system. This was done with the intention of substituting in real, collected RSSI signal strengths from the Bluetooth beacons once that stage is completed. The simulation allowed for the team to save much time in the process of developing such a system. The code is visible under section *2.2 Indoor Navigation Design*. The simulation is explained in detail under that section and it serves as the backbone for the navigation system. It simply requires the Raspberry Pi car motor actuation in order to move the car forward and perform turns.

## 2.5 Bill of Materials

| ITEM NO. | Part | QTY. |
|---|---|---|
| 1 | Raspberry Pi | 1 |
| 2 | ESP32 | 5 |
| 3 | Raspberry Pi Car Kit | 1 |
| 4 | Battery Pack for ESP32 | 4 |
| 5 | Power Brick for Raspberry Pi | 1 |
| 6 | Micro USB to USB type A cable | 1 |
| 7 | MicroSD card | 1 |
| 8 | Tablet | 1 |
| 9 | Omnidirectional Antenna w/SMA | 1 |
| 10 | Directional Antenna w/SMA | 4 |
| 11 | ESP32 Bluetooth Beacon Mont (maneuverable) | 4 |

## 2.6 Implementation Plan

Implementation of the ESP32 Bluetooth beacon system was carried out in phases. The steps taken to implement the design involves making diagrams to analyze during team brainstorming sessions, ordering and programing the ESP32s, testing to ensure the signal strengths are located and isolated, and implementing the final system with the autonomous navigation of the Raspberry Pi car. Currently, the RSSI signal strength of all four EPS32 Bluetooth beacons have been isolated in Node-RED. The signals now need to be integrated with the navigation simulation in order to actuate the car's movement.

*ESP32 Implementation*

The four ESP32 Bluetooth beacons which advertise themselves are uploaded with Bluetooth Low Energy server code through Arduino IDE. The ESP32 which is plugged into the Raspberry Pi and serves as a modem is uploaded with Bluetooth Low Energy scan code. The ESP32 plugged into the Raspberry Pi is to be connected with an omnidirectional SMA antenna. The four ESP32s which advertise themselves are to be connected with directional SMA antennas. The purpose of these antennas is to eliminate signal interference and ensure such a system is able to operate in relatively large-scale indoor environments.

*Figure 8 RSSI Signal Strength Mapping with Advertising Bluetooth Speaker*

| Distance (m) | Signal Strength Avg (dB) | Signal Strength Min, Max (c | 0.5 m | 1 m | 1.5 m | 2 m | 2.5 m | 3 m | | 0 m | | | 5.5 m |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0.5 | 47.15 | 45, 50 (Delta = 5) | 46 | 50 | 57 | 64 | 66 | 63 | | 30 | | | 70 |
| 1 | 53.85 | 50, 58 (Delta = 8) | 47 | 52 | 59 | 60 | 57 | 62 | | 30 | | | 70 |
| 1.5 | 58.35 | 54, 68 (Delta = 14) | 46 | 50 | 58 | 63 | 71 | 64 | | 29 | | | 78 |
| 2 | 60.6 | 59, 64 (Delta = 5) | 49 | 50 | 59 | 61 | 60 | 67 | | 29 | | | 67 |
| 2.5 | 61.8 | 56, 71 (Delta = 15) | 46 | 57 | 68 | 61 | 57 | 61 | | 29 | | | 71 |
| 3 | 62.75 | 60, 67 (Delta = 7) | 46 | 55 | 55 | 63 | 60 | 60 | | 30 | | | 71 |
| | | | 48 | 57 | 55 | 61 | 57 | 63 | | 30 | | | 69 |
| | | | 48 | 58 | 55 | 59 | 56 | 60 | | 30 | | | 67 |
| | | | 50 | 57 | 67 | 61 | 61 | 64 | | 30 | | | 67 |
| | | | 50 | 58 | 55 | 59 | 68 | 61 | | 30 | | | 78 |
| 5.5 | 69.45 | 65, 78 (Delta = 13) | 45 | 51 | 66 | 60 | 56 | 60 | | 30 | | | 71 |
| 0 | 29.75 | 28, 31 (Delta = 3) | 49 | 58 | 65 | 61 | 69 | 63 | | 29 | | | 72 |
| | | | 49 | 51 | 58 | 60 | 69 | 62 | | 31 | | | 65 |
| | | | 45 | 53 | 55 | 59 | 57 | 64 | | 30 | | | 67 |
| | | | 47 | 51 | 58 | 61 | 69 | 63 | | 30 | | | 66 |
| | | | 47 | 53 | 55 | 59 | 61 | 63 | | 30 | | | 72 |
| | | | 46 | 57 | 55 | 60 | 61 | 64 | | 28 | | | 66 |
| | | | 46 | 51 | 58 | 60 | 57 | 63 | | 29 | | | 67 |
| | | | 47 | 53 | 55 | 61 | 61 | 63 | | 30 | | | 66 |
| | | | 46 | 55 | 54 | 59 | 63 | 65 | | 31 | | | 69 |
| | | **Average** | 47.15 | 53.85 | 58.35 | 60.6 | 61.8 | 62.75 | | 29.75 | | | 69.45 |
| | | **Minimum** | 45 | 50 | 54 | 59 | 56 | 60 | | 28 | | | 65 |
| | | **Maxmimum** | 50 | 58 | 68 | 64 | 71 | 67 | | 31 | | | 78 |

*Figure 9 RSSI Signal Strength Mapping from Bluetooth Speaker Measurements and Raspberry Pi Signal Recordings*

As can be seen in *Figure 8*, an advertising Bluetooth Speaker was used in accordance with an ESP32 (uploaded with Bluetooth Low Energy scan code) plugged into a Raspberry Pi in order to map the RSSI signal strength. The intention of this mapping was to create a formula from which distance can be inferred once a signal strength is obtained. Twenty signal strengths were recorded at each distance in order to obtain an average as well as peak and trough values.

## 2.7 Rules and Regulations

There were restrictions that had to be satisfied when designing the automated indoor navigation system. The primary concern regarding the client’s demands was to develop a system which is adaptable to a variety of environments with ease. The simple way of approaching a navigation system is to use a tape-tracking system — however such a system is very rigorous to set up and does not offer the advantages that a Bluetooth navigation system does.

## 2.8 Intellectual Property

The George Brown College team required an automated indoor navigation system as per the request of the client, QRS. As this process was carried out through the Capstone course, it is important to note that the intellectual property of the design belongs to George Brown College.

## 2.9 Environmental Analysis

*Components Acquisition*

Our final product is an assembly of a pre-made robotic kit, several micro-controllers, cabling, and wood to create the stands for the beacons.

The environmental impact of the electronic components sourced would be the extraction of mineral ores to obtain rare earth minerals, iron, lead, copper and aluminum (board circuitry, batteries, IC chips, oscillator) , as well as the use of hydrocarbons to product plastics and

rubbers (used for the robot frame, tires, the encapsulation of the motors, wire insulation, battery and raspberry Pi cases, and switches).
All of these processes are energy intensive, and emit greenhouse gases, and the mineral extraction processes also produce waste-chemicals that should be contained and not released into the environment by the mines.
There are also emissions incurred by the logistics of transporting all these materials to the manufacturing facilities, and then transporting these items to distribution centers, and finally make its way to us through our main supplier, DigiKey.

*Assembly Process*

The assembly of the robotic kit did not require any further energy, materials, or manufacturing. Only human energy under the form of labor was expended.
The assembly of the micro-controllers and their battery cases however did require soldering, which emitted a negligible quantity of fumes. Lead-free solder was used, with a rosin core.

*Operation*

The operation of the robot, and the control methods (ie tablet) will all use electrical energy, stored inside battery packs. No emissions will result from the operation of the machine. Emissions may have resulted from the electricity used to charge said batteries, however.

*Maintenance and Repair*

Little to no maintenance is required for the robot assembly or the micro-controller beacons. The only requirement is to store the assembly in an environment sheltered from extreme temperature fluctuations and from the elements, as rain/snow may compromise the electronics via oxidation.

After a very prolonged use, it may be necessary to add small amounts of oil/grease in the gears of the motors and various moving parts. Most components however are manufactured to not require such interventions as the item life may be shorter than the actual life of the lubrication.

Repairing parts will be limited, and realistically the team will resort to swapping defective components, such as the servo-motors, or a micro-controller in case it is found to be defective, damaged, or somehow fails operation. Each component that will be replaced will come with an extra emissions cost, for production and logistics.

*End of Life Disposal*

Once the robotic assembly will no longer be functional, due to the malfunctioning of the onboard raspberry pi, or due to it reaching its useful life range, it will have to be disassembled into its basic components. The plastic body can be re-utilized for another project or discarded as garbage. All the hardware can be re-used or recycled as metals. The biggest environmental concern will be the batteries used, if not properly recycled or disposed of in an approved manner. There may be some materials eligible for recycling, like the wiring, or the rare earth metals in the micro-controllers, or the surface mount electronic components. Post-secondary institutions present E-Waste facilities that may take in the assemblies free of charge, in order to assist with environmental efforts. Recycling however may not result to be energy efficient due to the small amount of materials present compared to the energy required (and emissions) to recycle them.

## 2.10 Social Analysis

Technologies that provide an automated way of navigating through areas are becoming more mainstream and may greatly aid those that may have difficulty doing so: people with disabilities, the elderly, or even lost children, may all benefit from this type of navigation system. In fact, the technology can be integrated into a “building guide” that will be able to guide you to a certain store withing a mall, for example, or to find a specific service within a government building. The ubiquitous nature of using Bluetooth and Wifi networks would make the deployment of this system easily attenable in a variety of settings.

## 2.11 Economic Analysis

The main costs associated with the project are from purchasing the electronic components and the robotic assembly kit. A small portion of that cost is for shipping/handling charges to receive the items. Due to the team performing the assembling and soldering, no third-party labor costs were incurred. The battery-cases came with the wiring, and soldering materials and tools were already in possession by the team, as well as the tablet-PC used to control the robot. The project incurred a final total cost of 243.56$

# 3.0 Future Work and Conclusions

In the future, the automated indoor navigation system can be altered in order to compensate for large-scale tasks which can be performed by the Raspberry Pi car. Should the client wish to pursue the proof of concept and desire a system to perform tasks, the Raspberry Pi car would need to be larger and more capable of producing a large amount of torque. Additionally, features such as robotic arms or hooks can be designed and installed in order to provide a task-capable robot. Furthermore, a voice-navigation system could be implemented with the Raspberry Pi car in order to allow for auditory commands. Such a feature is even more beneficial in light of the pandemic. Should the client wish to implement the automated indoor navigation system in large environments such as warehouses, more research should be done in signal boosting and reducing interference to ensure the navigation works as necessary.

# Reference List

Electronnicproject. (August 29, 2017). Wireless Signal Meter App for Raspberry Pi Using Node-Red. *Electron Nic Project*. Retrieved from: https://electronnicproject.blogspot.com/2017/08/wireless-signal-meter-app-for-raspberry.html

Github. (n.d.). RaspberryPi-Signal-Meter. *Github*. Retrieved from: https://github.com/m2msupport/RaspberryPi-Signal-Meter

Github. (November 29, 2017). How to detect RSSI from BLE on ESP32?. *Github*. Retrieved from: https://github.com/nkolban/esp32-snippets/issues/236

Meraki. (June 11, 2018). Iphones and Androids Bluetooth not showing on BLE scanning/clients. *Meraki*. Retrieved from: https://community.meraki.com/t5/Wireless-LAN/Iphones-and-Androids-Bluetooth-not-showing-on-BLE-scanning/td-p/21170

Nodered. (August 18, 2018). Searching character’s. *Nodered*. Retrieved from: https://discourse.nodered.org/t/searching-characters/2348/8

# Appendix A – Node-RED Screenshots

This appendix contains the Node-RED Flow and Dashboard progression.

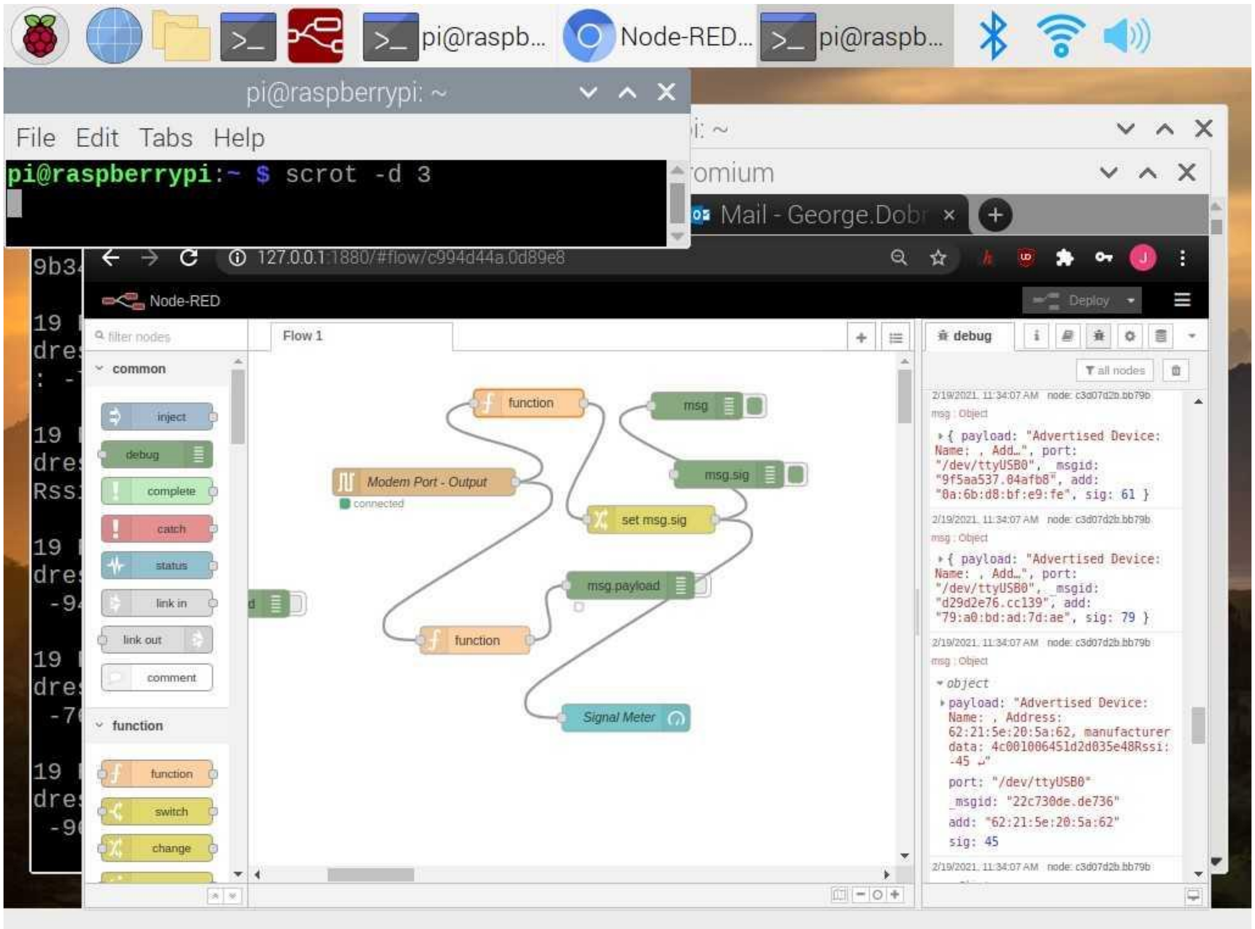



*Figure 10Early Node-RED flow which outputs raw, unformatted strings of Bluetooth Advertised Devices containing device IDs and RSSI signal strengths (the latter of which was isolated using a function node, and subsequently converted into a number)*

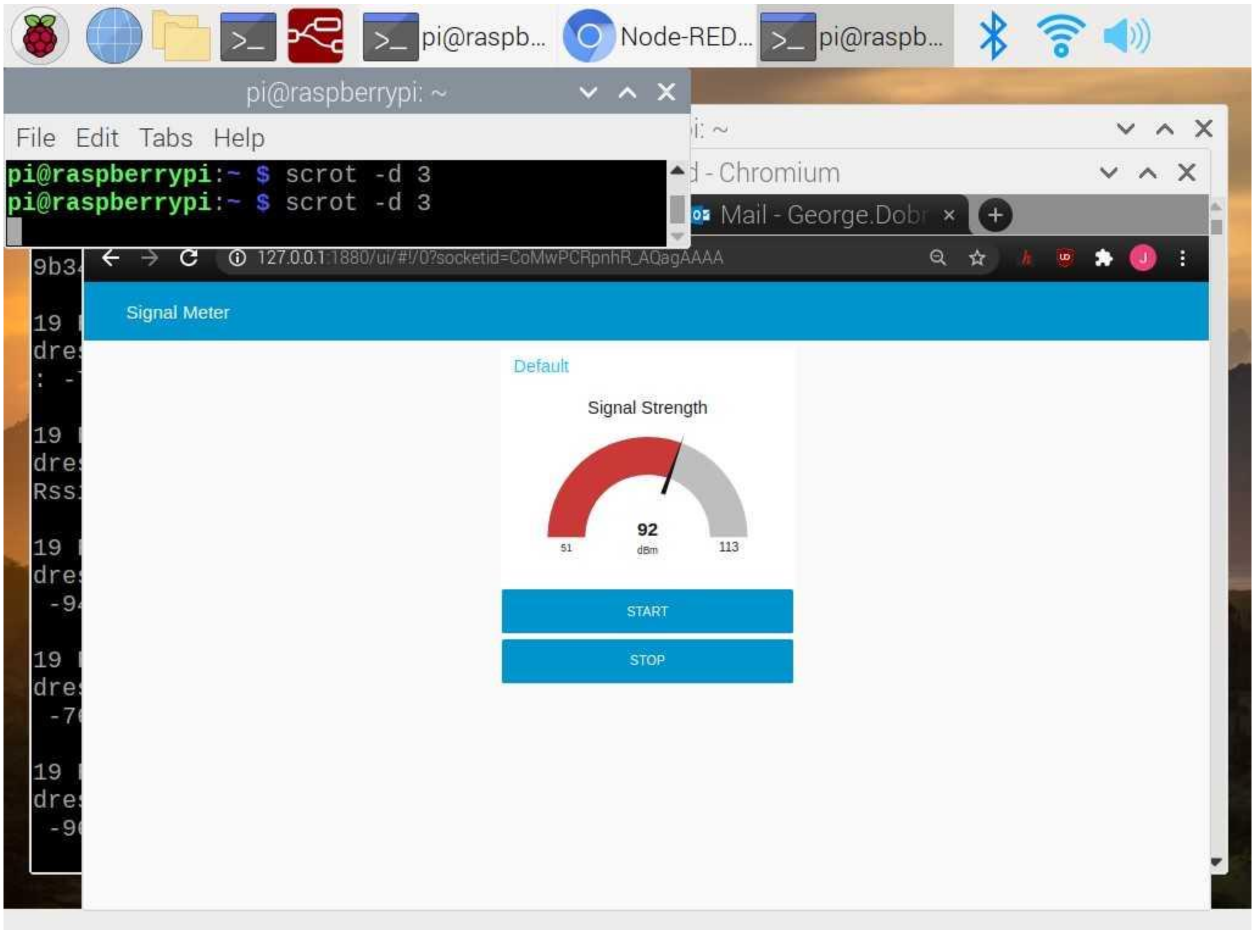



*Figure 11 Gauge displaying the signal strength of all cycled devices on Node-RED Dashboard with start and stop buttons that control the Bluetooth scanning.*

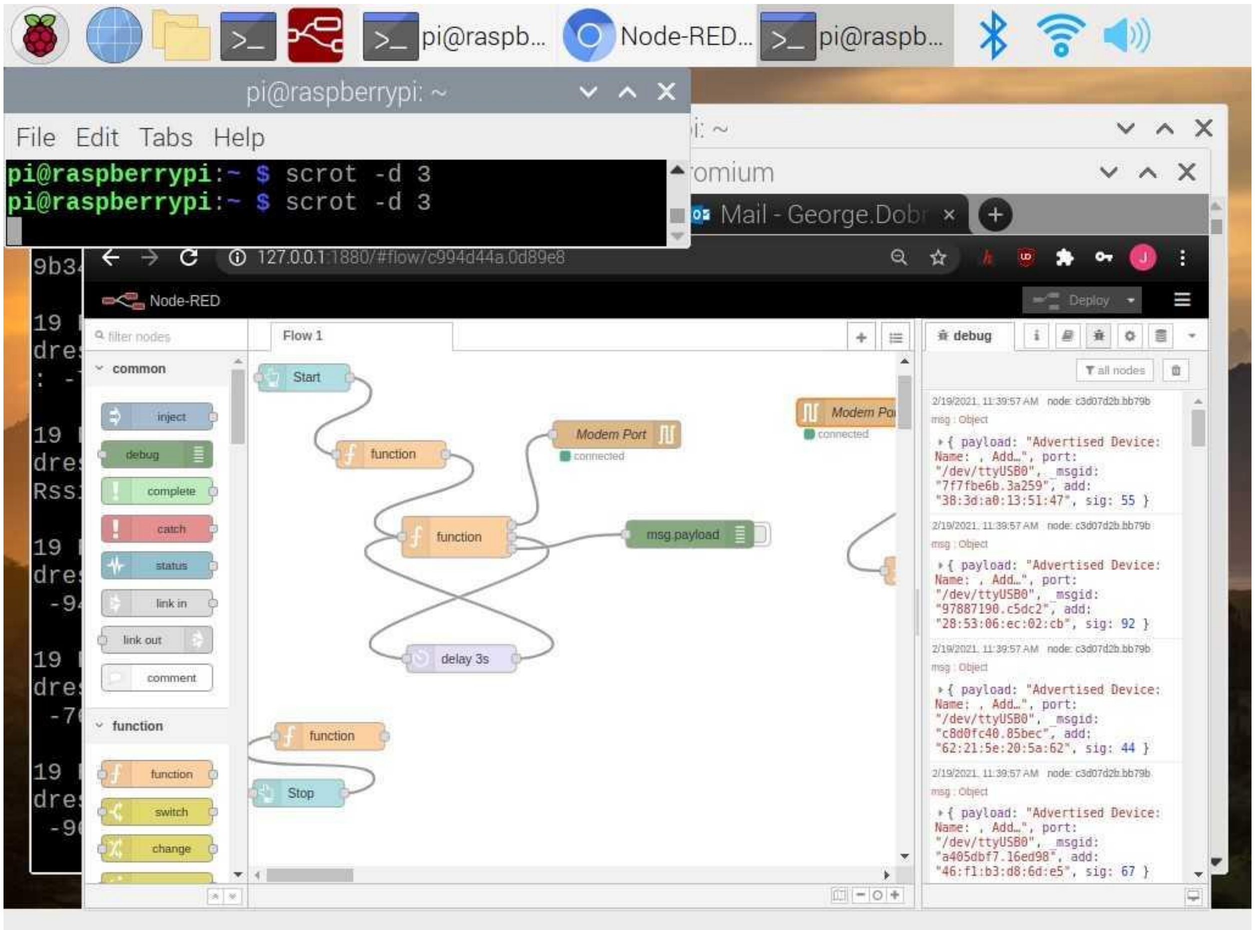



*Figure 12 The left side of the flow which initiates the Bluetooth device scan if the start button is pressed on Node-RED dashboard. There is a Stop button node at the bottom which stops the scan.*

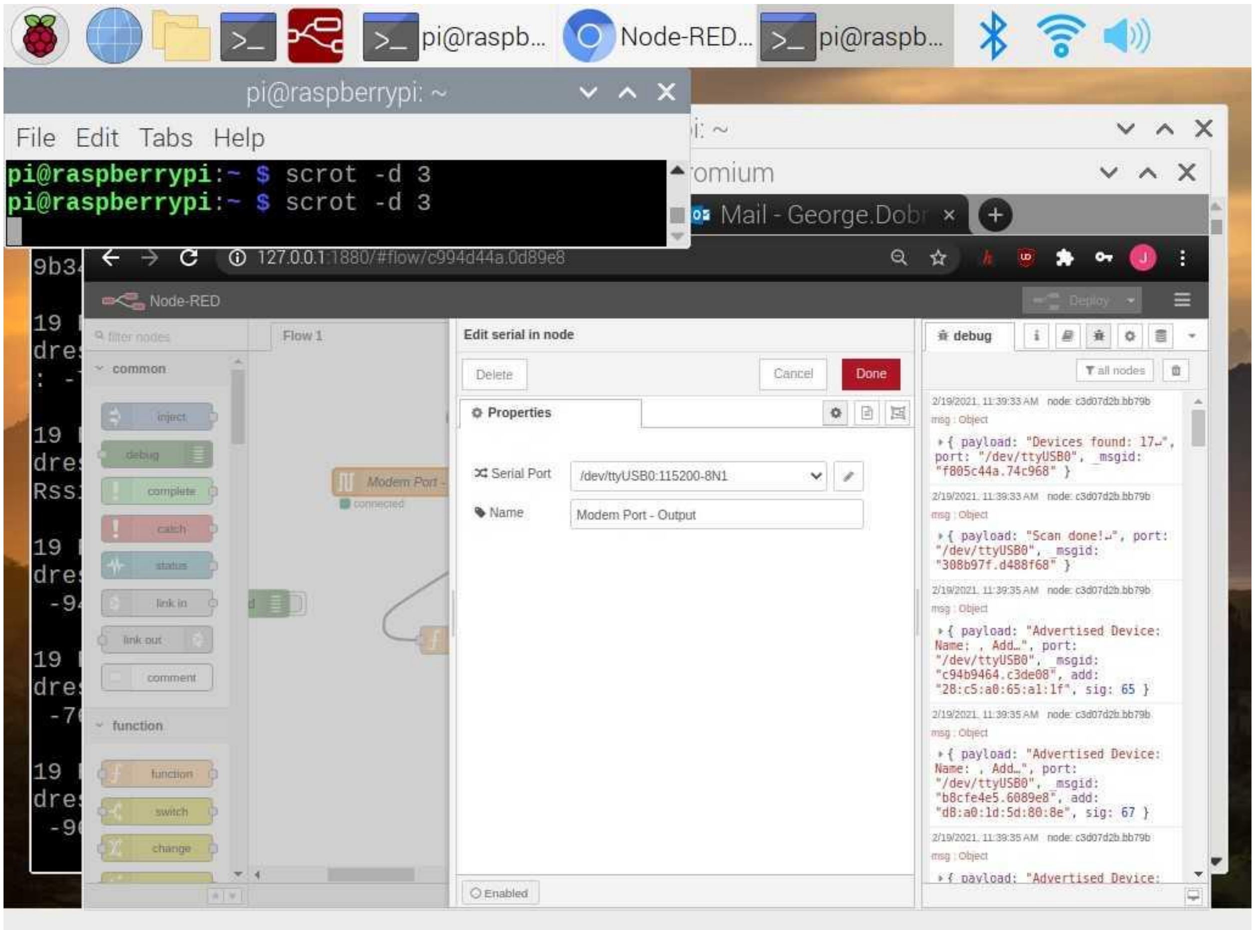



*Figure 13 The serial node which is used to establish a connection between the ESP32 plugged into the Raspberry Pi and the Raspberry Pi itself. The EPS32 is programmed to run a BLE scan.*

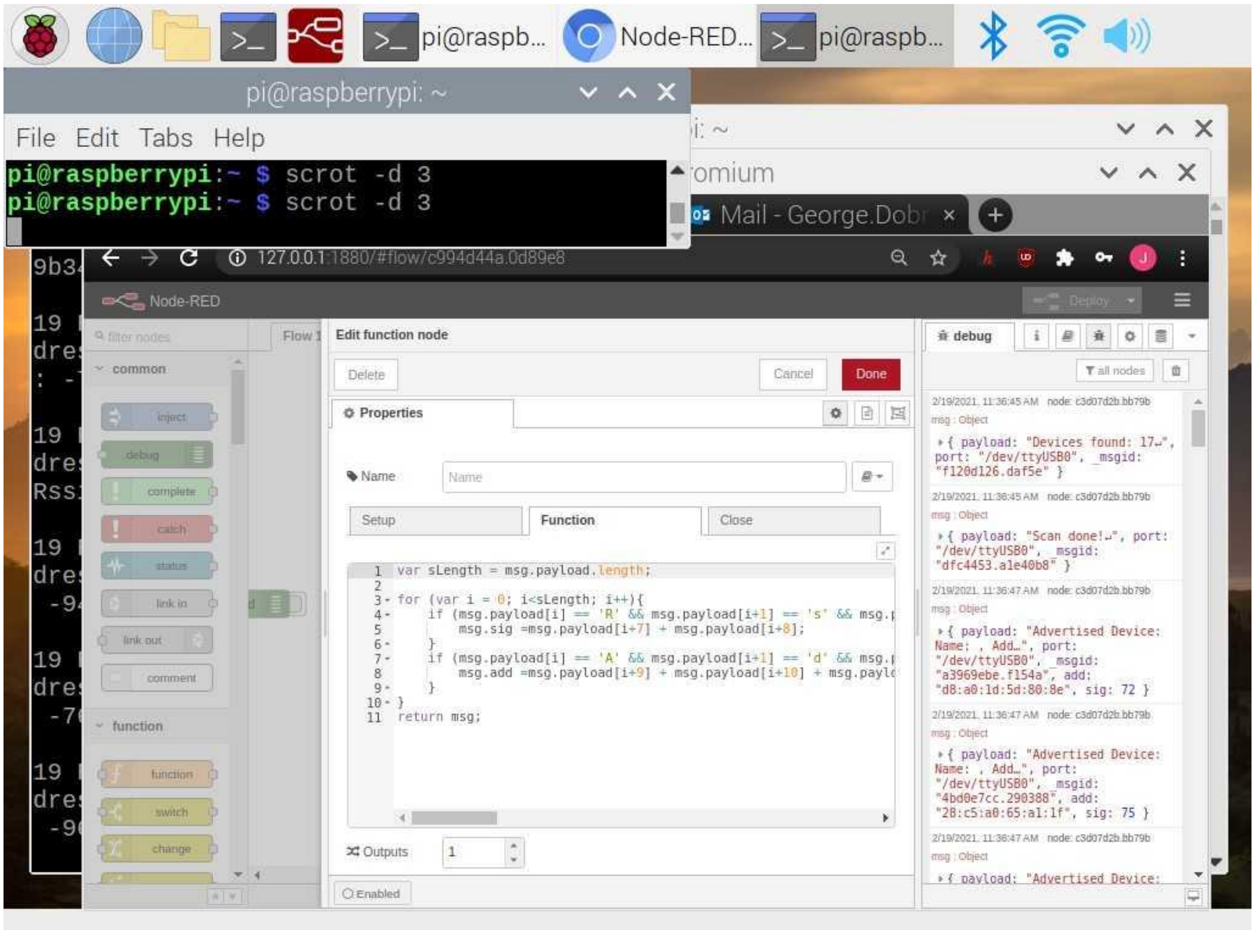



*Figure 14 The function node on the right side of the flow is shown here which analyses the raw, unformatted string and isolates each devices' address and RSSI signal strength by searching through the raw strings.*

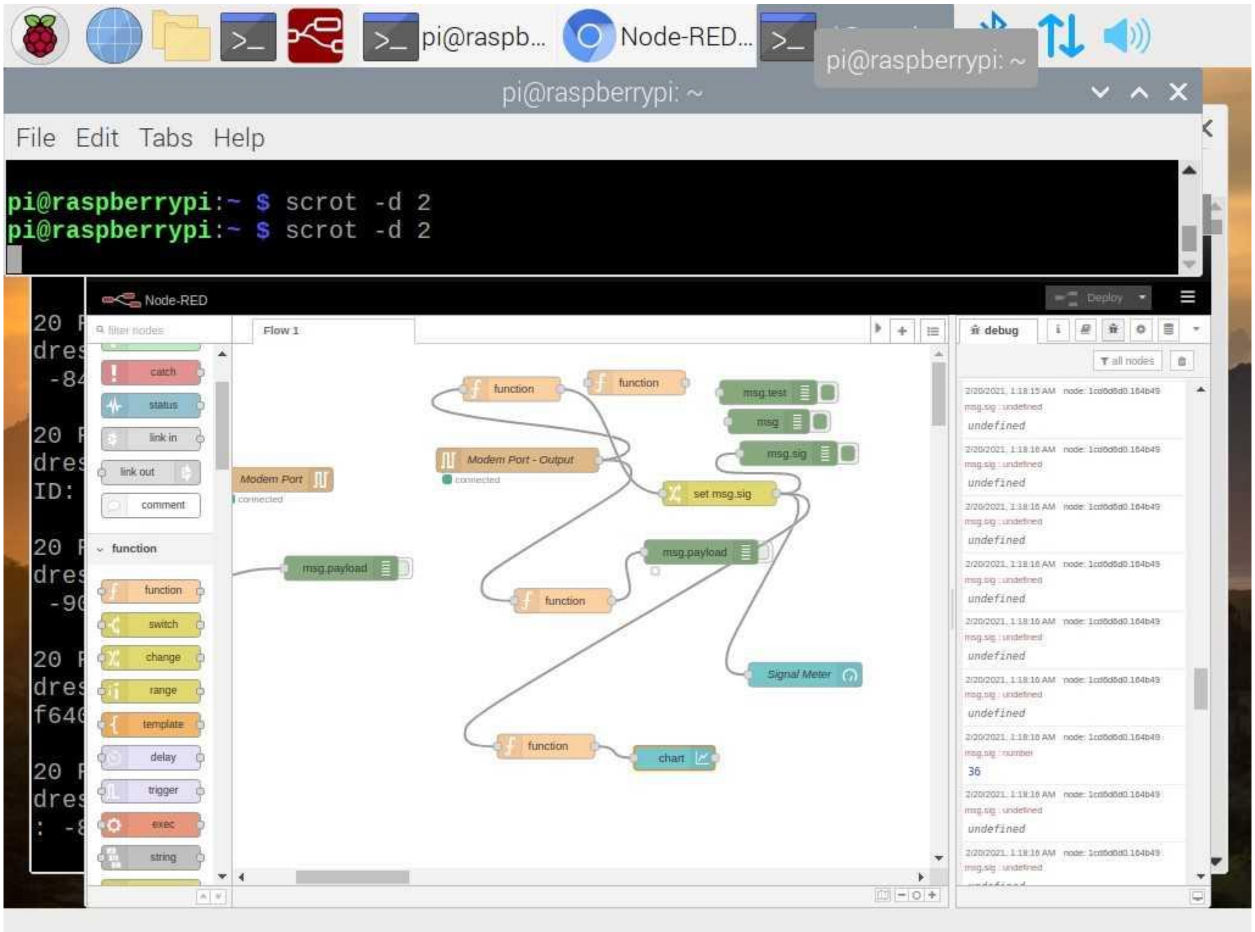



*Figure 15 The debug log shows one signal being isolated, while the other signals are being ignored. This isolation is cleaned up to only display the isolated signal in one of the following Node-RED screenshots.*

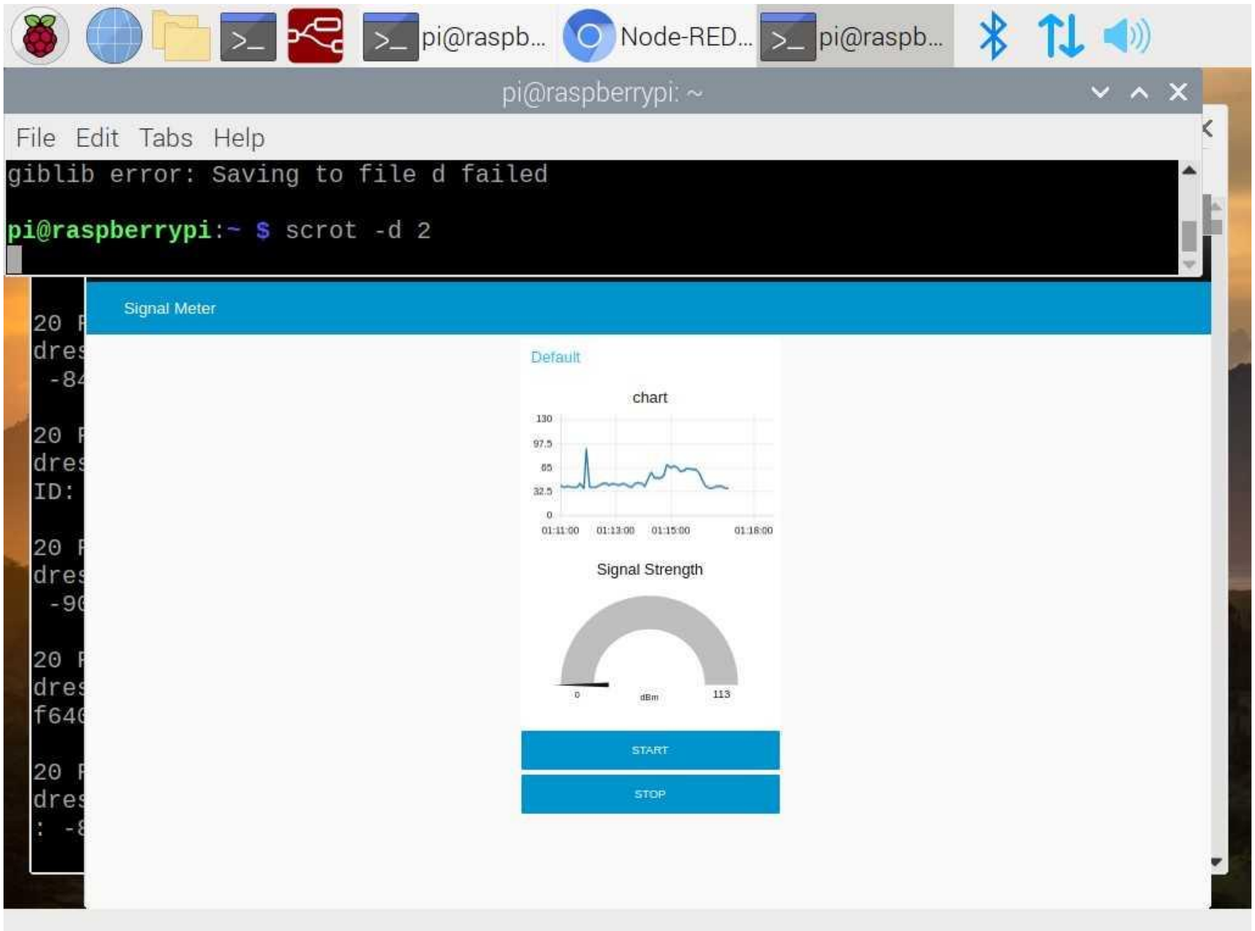



*Figure 16 This is the Node-RED dashboard which contains a graph which charts a single, isolated RSSI signal strength over time. The graph shows an increase and decrease; this was done by moving the Bluetooth device away form the Raspberry Pi, and then back.*

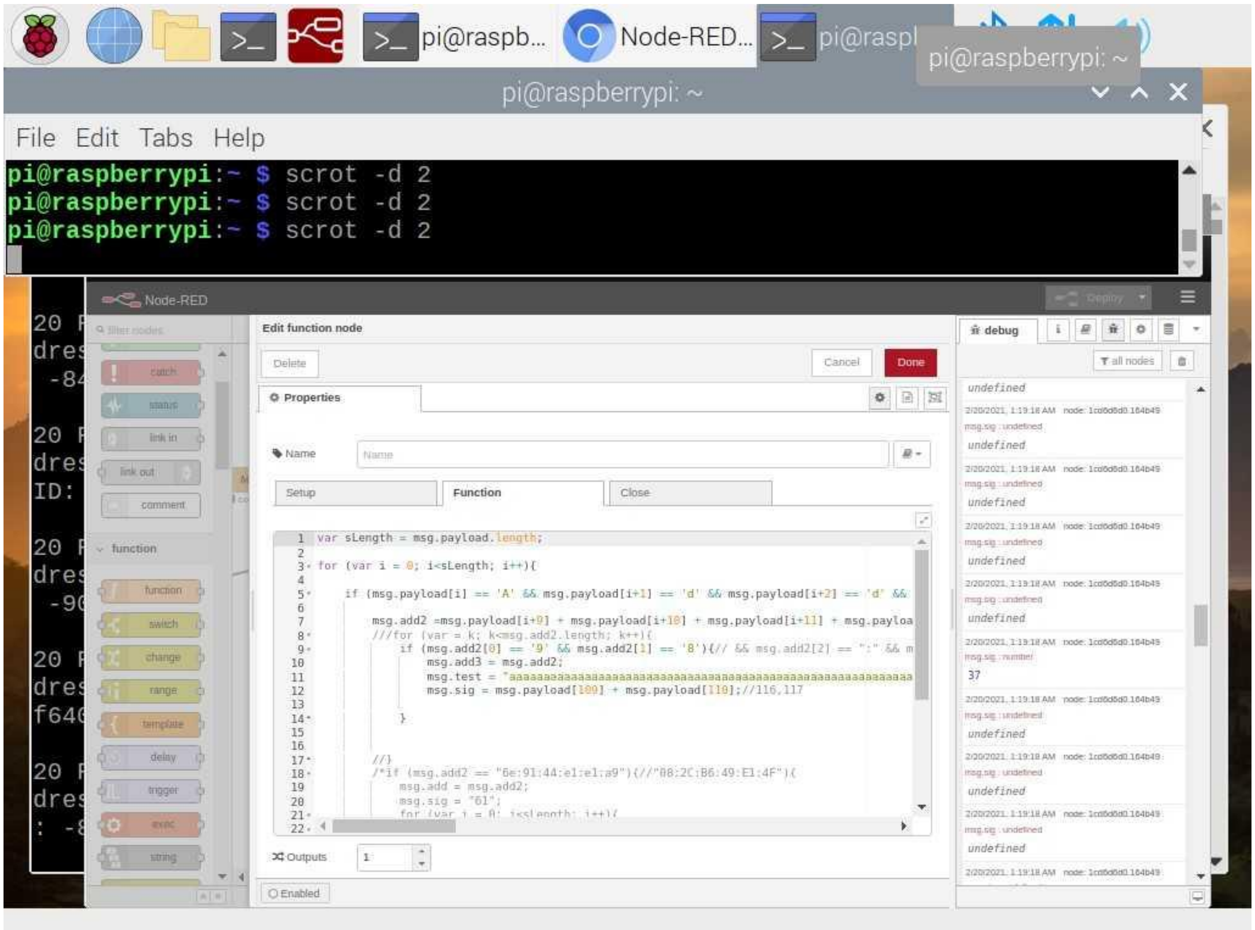



*Figure 17 This is the function node previously shown but adjusted in order to isolate the address and RSSI signal strength of a single desired device. This will be used to isolate beacon signals only in the future.*

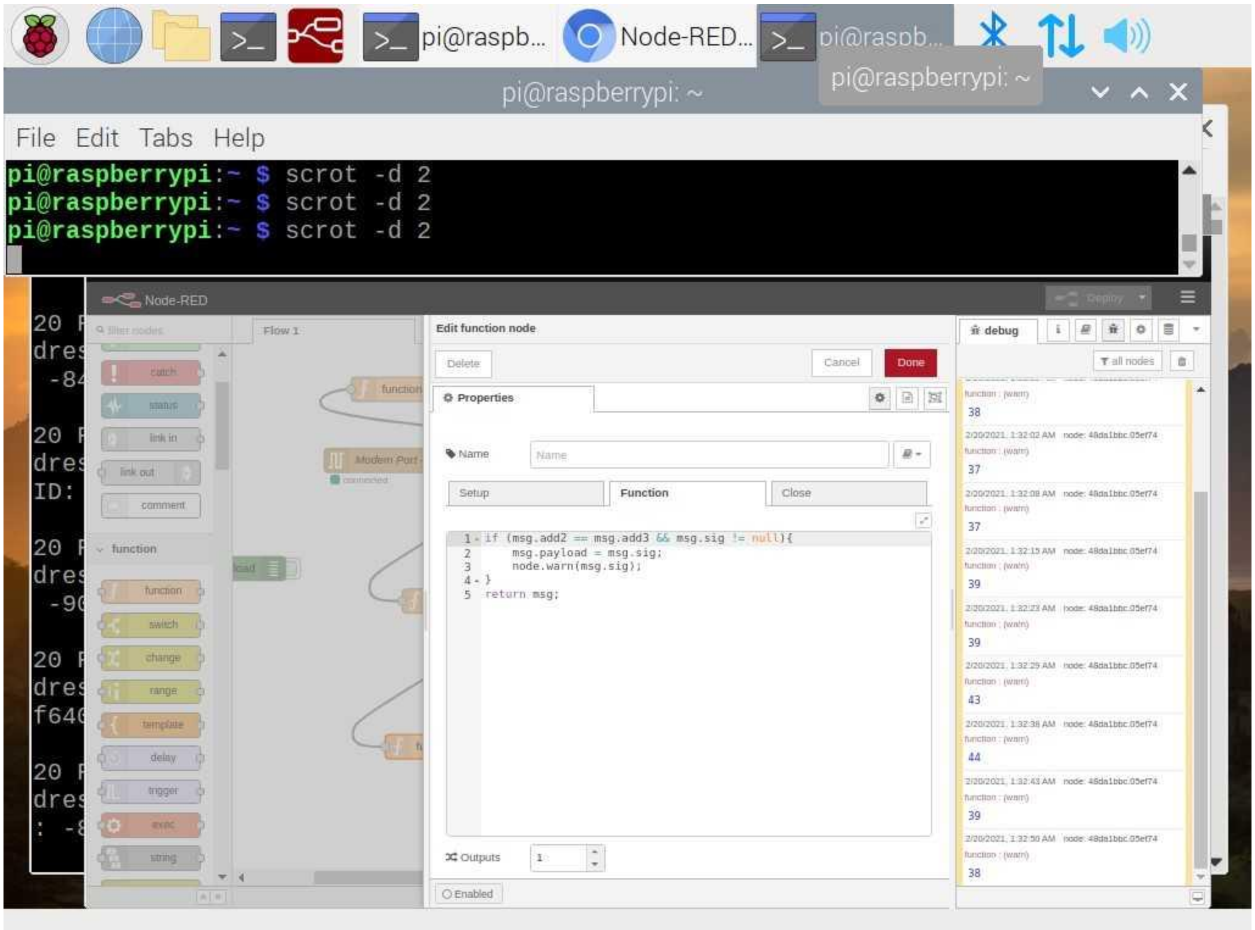



*Figure 18 This is a new function node which was created to isolate one the single RSSI signal strength in order to avoid unnecessary spam in the debug log and to ensure proper operation in general.*

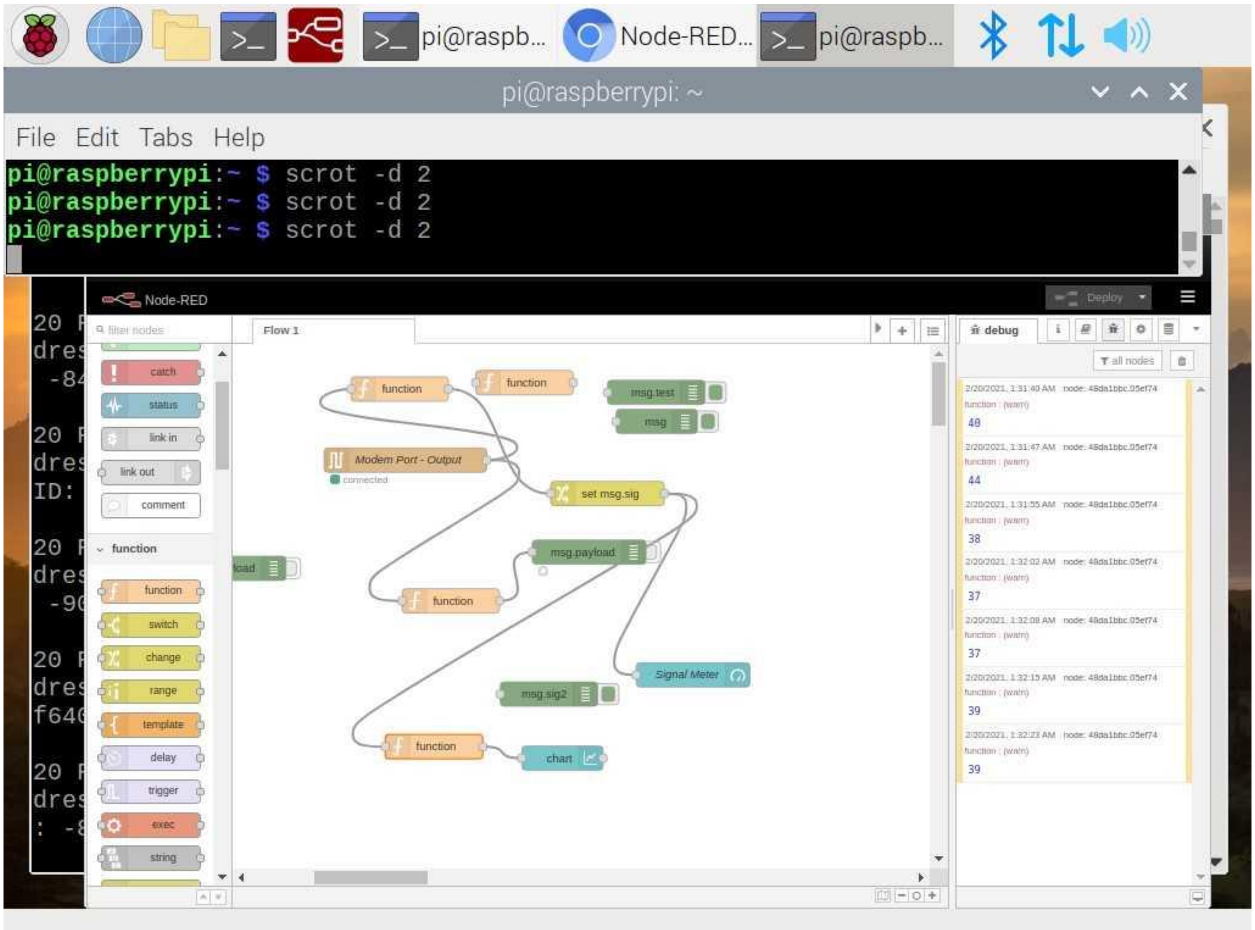



*Figure 19 This shows the updated, right side of the flow which includes the Figure 9 node and chart node.*

# Appendix B – Node-RED Exported Flow

This Appendix contains the Node-RED Flow export. The flow is tailored to isolate the Bluetooth signal from four ESP32 devices which have been set up as servers. A function searches through each device's raw, unformatted string and isolates the four necessary RSSI signal strength values. Moreover, there is a function node which simulates automated indoor navigation. This node provides the backbone for the navigation system and simply requires the plugging in of RSSI signal strength values and the Raspberry Pi car motion actuation.

[{"id":"c994d44a.0d89e8","type":"tab","label":"Flow
1","disabled":false,"info":""},{"id":"e967b861.9545b8","type":"ui_gauge","z":"c994d44a.0d89e8","name
":"Signal
Meter","group":"910f87cf.9376b8","order":0,"width":0,"height":0,"gtype":"gage","title":"Signal
Strength","label":"dBm","format":"{{msg.sig}}","min":"0","max":"113","colors":["#00b500","#e6e600","
#ca3838"],"seg1":"-85","seg2":"-
65","x":1054.0000686645508,"y":407.00001335144043,"wires":[]},{"id":"c570e1b8.88a71","type":"func
tion","z":"c994d44a.0d89e8","name":"","func":"var rssi_stat = global.get(\"get_rssi\");\n\nif(rssi_stat
=== 0)\n{\n    msg.payload = \"Stopping RSSI Reads from
modem\";\n    \n    return[null,null,msg];\n}\nelse if(rssi_stat ===1)\n{\n  msg.payload = \"AT+CSQ\" +
\"\\r\\n\";\n  return[msg,msg,msg];\n    \n}\n","outputs":"3","noerr":0,"initialize":"","finalize":"","x":22
8,"y":279,"wires":[["7cfad5e2.5fcf7c"],["9a609fd8.09d7c"],["1c1e2998.ef6f96"]]},{"id":"7cfad5e2.5fcf7c"
,"type":"serial out","z":"c994d44a.0d89e8","name":"Modem
Port","serial":"93485f7d.0073f","x":403,"y":167,"wires":[]},{"id":"5f9d9228.2f6c0c","type":"serial
in","z":"c994d44a.0d89e8","name":"Modem Port -
Output","serial":"93485f7d.0073f","x":700,"y":143,"wires":[["efb5c023.7557c","9a0eaf0b.cdcbd"]]},{"id
":"31adc3e0.f85f9c","type":"debug","z":"c994d44a.0d89e8","name":"","active":false,"tosidebar":true,"c
onsole":false,"tostatus":true,"complete":"payload","targetType":"msg","statusVal":"payload","statusTy
pe":"auto","x":926.0000247955322,"y":256.0000123977661,"wires":[]},{"id":"9a609fd8.09d7c","type":"
delay","z":"c994d44a.0d89e8","name":"","pauseType":"delay","timeout":"3","timeoutUnits":"seconds",
"rate":"1","nbRateUnits":"1","rateUnits":"second","randomFirst":"1","randomLast":"5","randomUnits":
"seconds","drop":false,"x":233,"y":412,"wires":[["c570e1b8.88a71"]]},{"id":"1c1e2998.ef6f96","type":"d
ebug","z":"c994d44a.0d89e8","name":"","active":false,"tosidebar":true,"console":false,"tostatus":false,
"complete":"payload","targetType":"msg","statusVal":"","statusType":"auto","x":483,"y":276,"wires":[]}
,{"id":"f794922.2f60b7","type":"ui_button","z":"c994d44a.0d89e8","name":"","group":"910f87cf.9376b
8","order":0,"width":0,"height":0,"passthru":false,"label":"Start","color":"","bgcolor":"","icon":"","paylo
ad":"","payloadType":"str","topic":"","x":61,"y":105,"wires":[["cc1ae55b.9c5e28"]]},{"id":"cc1ae55b.9c5
e28","type":"function","z":"c994d44a.0d89e8","name":"","func":"global.set(\"get_rssi\",1);\nreturn
msg;","outputs":1,"noerr":0,"initialize":"","finalize":"","x":156,"y":189,"wires":[["c570e1b8.88a71"]]},{"i
d":"8c7080e8.3ce1e","type":"ui_button","z":"c994d44a.0d89e8","name":"","group":"910f87cf.9376b8",
"order":0,"width":0,"height":0,"passthru":false,"label":"Stop","color":"","bgcolor":"","icon":"","payload"
:"","payloadType":"str","topic":"","x":55,"y":559,"wires":[["6bf040e1.3b0be"]]},{"id":"6bf040e1.3b0be",
"type":"function","z":"c994d44a.0d89e8","name":"","func":"global.set(\"get_rssi\",0);\nreturn
msg;","outputs":1,"noerr":0,"x":89,"y":497,"wires":[[]]},{"id":"efb5c023.7557c","type":"function","z":"c9
94d44a.0d89e8","name":"","func":"var temp;\nvar temp2,dbm,rssi;\n\nvar mystr =
msg.payload;\nnode.log(mystr);\nif(mystr.indexOf(\"+CSQ:\")>-1)\n{\ntemp = mystr.split(\"+CSQ:

\");\ntemp[0] = temp[0].trim();\nnode.log(temp[1]);\ntemp2 = temp[1].split(\",\");\nrssi = temp2[0];\n//node.log(\"Value of temp[0], temp2[0]\", temp[0], temp2[0]);\ndbm = -113 + (2*rssi);\n\nmsg.payload = dbm;\n\nreturn msg;\n}\nelse\n return null;\n\n\n","outputs":1,"noerr":0,"initialize":"","finalize":"","x":756,"y":316,"wires":[["31adc3e0.f85f9c"]]},{"id":"1cd6d6d0.164b49","type":"debug","z":"c994d44a.0d89e8","name":"","active":true,"tosidebar":true,"console":false,"tostatus":false,"complete":"sig2","targetType":"msg","statusVal":"","statusType":"auto","x":876.6667823791504,"y":430.6667003631592,"wires":[]},{"id":"69198a60.2075b4","type":"function","z":"c994d44a.0d89e8","name":"","func":"var sLength = msg.payload.length;\n\nfor (var i = 0; i<sLength; i++){\n\n if (msg.payload[i] == 'U' && msg.payload[i+1] == 'U' && msg.payload[i+2] == 'I' && msg.payload[i+3] == 'D'){ //&& msg.payload[i+4] == 'e' && msg.payload[i+5] == 's' && msg.payload[i+6] == 's'){\n \n msg.add2 =msg.payload[i+6] + msg.payload[i+7] + msg.payload[i+8] + msg.payload[i+9]; //+ msg.payload[i+13] + msg.payload[i+14] + msg.payload[i+15] + msg.payload[i+16] + msg.payload[i+17] + msg.payload[i+18] + msg.payload[i+19] + msg.payload[i+20] + msg.payload[i+21] + msg.payload[i+22] + msg.payload[i+23] + msg.payload[i+24] + msg.payload[i+25];\n ///for (var = k; k<msg.add2.length; k++){\n if (msg.add2[0] == '5' && msg.add2[1] == 'a' && msg.add2[2] == \"5\" && msg.add2[3] == 'b'){ //&& msg.add2[4] == 'c' && msg.add2[5] == \":\" && msg.add2[6] == 'b' && msg.add2[7] == '6' && msg.add2[8] == \":\" && msg.add2[9] == '4' && msg.add2[10] == '9' && msg.add2[11] == \":\" && msg.add2[12] == 'e' && msg.add2[13] == '1' && msg.add2[14] == \":\" && msg.add2[15] == '4' && msg.add2[16] == 'f' ){\n msg.test = \"aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa\";\n msg.sig = msg.payload[111] + msg.payload[112];
//116,117\n \n }\n \n \n //}\n /*if (msg.add2 == \"6e:91:44:e1:e1:a9\"){//\"08:2C:B6:49:E1:4F\"){\n msg.add = msg.add2;\n msg.sig = \"61\";\n for (var j = 0; j<sLength; j++){\n if (msg.payload[j] == 'R' && msg.payload[j+1] == 's' && msg.payload[j+2] == 's' && msg.payload[j+3] == 'i'){\n msg.sig = \"52\";//msg.payload[j+7] + msg.payload[j+8];\n \n //msg.payload = \"aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa\";\n }\n }\n }* /\n }\n }\n \n \n\nreturn msg;","outputs":1,"noerr":0,"initialize":"","finalize":"","x":846.6667556762695,"y":47.66667938232422,"wires":[[]]},{"id":"c3d07d2b.bb79b","type":"debug","z":"c994d44a.0d89e8","name":"","active":true,"tosidebar":true,"console":false,"tostatus":false,"complete":"test","targetType":"msg","statusVal":"","statusType":"auto","x":1008.6666870117188,"y":59.666664123535156,"wires":[]},{"id":"adec5088.b64db","type":"change","z":"c994d44a.0d89e8","name":"","rules":[{"t":"set","p":"sig","pt":"msg","to":"$number(sig)","tot":"jsonata"},{"t":"set","p":"sig2","pt":"msg","to":"$number(sig2)","tot":"jsonata"},{"t":"set","p":"sig3","pt":"msg","to":"$number(sig3)","tot":"jsonata"},{"t":"set","p":"sig4","pt":"msg","to":"$number(sig4)","tot":"jsonata"}],"action":"","property":"","from":"","to":"","reg":false,"x":958.6666870117188,"y":184.0000171661377,"wires":[["e967b861.9545b8","48da1bbc.05ef74"]]},{"id":"9a0eaf0b.cdcbd","type":"function","z":"c994d44a.0d89e8","name":"","func":"var sLength = msg.payload.length;\n\nfor (var i = 0; i<sLength; i++){\n\n if (msg.payload[i] == 'U' && msg.payload[i+1] == 'U' && msg.payload[i+2] == 'I' && msg.payload[i+3] == 'D'){// && msg.payload[i+4] == 'e' && msg.payload[i+5] == 's' && msg.payload[i+6] == 's'){\n \n msg.add2 =msg.payload[i+6] + msg.payload[i+7] + msg.payload[i+8] + msg.payload[i+9] + msg.payload[i+10] + msg.payload[i+11] + msg.payload[i+12] + msg.payload[i+13];// + msg.payload[i+17] + msg.payload[i+18] + msg.payload[i+19] + msg.payload[i+20] + msg.payload[i+21] + msg.payload[i+22] + msg.payload[i+23] + msg.payload[i+24] +

msg.payload[i+25];\n ///for (var = k; k<msg.add2.length; k++){\n if (msg.add2[0] == '4' && msg.add2[1] == 'f'){// && msg.add2[2] == \":\" && msg.add2[3] == '2' && msg.add2[4] == 'c' && msg.add2[5] == \":\" && msg.add2[6] == 'b' && msg.add2[7] == '6' && msg.add2[8] == \":\" && msg.add2[9] == '4' && msg.add2[10] == '9' && msg.add2[11] == \":\" && msg.add2[12] == 'e' && msg.add2[13] == '1' && msg.add2[14] == \":\" && msg.add2[15] == '4' && msg.add2[16] == 'f' ){\n msg.add3 = msg.add2;\n msg.test = \"aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa\";\n msg.sig = msg.payload[111] + msg.payload[112];//116,117\n \n }\n if (msg.add2[0] == '5' && msg.add2[1] == 'f'){// && msg.add2[2] == \":\" && msg.add2[3] == '2' && msg.add2[4] == 'c' && msg.add2[5] == \":\" && msg.add2[6] == 'b' && msg.add2[7] == '6' && msg.add2[8] == \":\" && msg.add2[9] == '4' && msg.add2[10] == '9' && msg.add2[11] == \":\" && msg.add2[12] == 'e' && msg.add2[13] == '1' && msg.add2[14] == \":\" && msg.add2[15] == '4' && msg.add2[16] == 'f' ){\n msg.add4 = msg.add2;\n msg.test = \"aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa\";\n msg.sig2 = msg.payload[111] + msg.payload[112];//116,117\n \n }\n if (msg.add2[0] == '6' && msg.add2[1] == 'f'){// && msg.add2[2] == \":\" && msg.add2[3] == '2' && msg.add2[4] == 'c' && msg.add2[5] == \":\" && msg.add2[6] == 'b' && msg.add2[7] == '6' && msg.add2[8] == \":\" && msg.add2[9] == '4' && msg.add2[10] == '9' && msg.add2[11] == \":\" && msg.add2[12] == 'e' && msg.add2[13] == '1' && msg.add2[14] == \":\" && msg.add2[15] == '4' && msg.add2[16] == 'f' ){\n msg.add5 = msg.add2;\n msg.test = \"aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa\";\n msg.sig3 = msg.payload[111] + msg.payload[112];//116,117\n \n } \n if (msg.add2[0] == '7' && msg.add2[1] == 'f'){// && msg.add2[2] == \":\" && msg.add2[3] == '2' && msg.add2[4] == 'c' && msg.add2[5] == \":\" && msg.add2[6] == 'b' && msg.add2[7] == '6' && msg.add2[8] == \":\" && msg.add2[9] == '4' && msg.add2[10] == '9' && msg.add2[11] == \":\" && msg.add2[12] == 'e' && msg.add2[13] == '1' && msg.add2[14] == \":\" && msg.add2[15] == '4' && msg.add2[16] == 'f' ){\n msg.add6 = msg.add2;\n msg.test = \"aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa\";\n msg.sig4 = msg.payload[111] + msg.payload[112];//116,117\n \n } \n //}\n /*if (msg.add2 == \"6e:91:44:e1:e1:a9\"){//\"08:2C:B6:49:E1:4F\"){\n msg.add = msg.add2;\n msg.sig = \"61\";\n for (var j = 0; j<sLength; j++){\n if (msg.payload[j] == 'R' && msg.payload[j+1] == 's' && msg.payload[j+2] == 's' && msg.payload[j+3] == 'i'){\n msg.sig = \"52\";//msg.payload[j+7] + msg.payload[j+8];\n \n //msg.payload = \"aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa\";\n }\n }\n }*/\n }\n }\n \n \n\nreturn msg;","outputs":1,"noerr":0,"initialize":"","finalize":"","x":693.0078506469727,"y":55.523444175720215,"wires":[["adec5088.b64db"]]},{"id":"bc6d6c1d.dc2e6","type":"debug","z":"c994d44a.0d89e8","name":"","active":true,"tosidebar":true,"console":false,"tostatus":false,"complete":"true","targetType":"full","statusVal":"","statusType":"auto","x":1009.2578125,"y":95.4453125,"wires":[]},{"id":"c316448b.1ad4c8","type":"ui_chart","z":"c994d44a.0d89e8","name":"","group":"910f87cf.9376b8","order":3,"width":0,"height":0,"label":"chart","chartType":"line","legend":"false","xformat":"HH:mm:ss","interpolate":"linear","nodata":"","dot":false,"ymin":"0","ymax":"130","removeOlder":1,"removeOlderPoints":"","removeOlderUnit":"3600","cutout":0,"useOneColor":false,"colors":["#1f77b4","#aec7e8","#ff7f0e","#2ca02c","#98df8a","#d62728","#ff9896","#9467bd","#c5b0d5"],"useOldStyle":false,"outputs":1,"x":893.2421875,"y

":509.3984375,"wires":[[]]},{"id":"48da1bbc.05ef74","type":"function","z":"c994d44a.0d89e8","name":"","func":"/*if (msg.avg == null)\n    msg.avg = 0;\n//if (msg.runs == null)\n//    msg.runs = 0;\n\nif (msg.add2 == msg.add3 && msg.sig != null){\n    msg.runs += 1;\n    msg.payload = msg.sig;\n    msg.avg = (msg.avg + msg.sig)/msg.runs;\n    node.warn(msg.runs);\n}\nreturn msg;*/\n\nif (msg.add2 == msg.add3 && msg.sig != null){\n    msg.payload = msg.sig;\n    node.warn(msg.sig);\n}\nif (msg.add2 == msg.add4 && msg.sig2 != null){\n    msg.payload = msg.sig2;\n    node.warn(msg.sig2);\n}\nif (msg.add2 == msg.add5 && msg.sig3 != null){\n    msg.payload = msg.sig3;\n    node.warn(msg.sig3);\n}\nif (msg.add2 == msg.add6 && msg.sig4 != null){\n    msg.payload = msg.sig4;\n    node.warn(msg.sig4);\n}\nreturn msg;","outputs":1,"noerr":0,"initialize":"","finalize":"","x":735.2421875,"y":495.2421875,"wires":[["c316448b.1ad4c8"]]},{"id":"86b04882.59f818","type":"average","z":"c994d44a.0d89e8","name":"","topic":"","x":1120.000072479248,"y":330.6666660308838,"wires":[[]]},{"id":"9acd67ae.5710b8","type":"change","z":"c994d44a.0d89e8","name":"","rules":[{"t":"set","p":"payload","pt":"msg","to":"sig","tot":"msg"}],"action":"","property":"","from":"","to":"","reg":false,"x":917.0000305175781,"y":462.3333511352539,"wires":[[]]},{"id":"3eb659bc.974586","type":"function","z":"c994d44a.0d89e8","name":"","func":"/*  A    B\n\n\n\n    C       D*/\nmsg.aX = 0;\nmsg.aY = 0;\nmsg.bX = 10;\nmsg.bY = 0;\nmsg.cX = 0;\nmsg.cY = 10;\nmsg.dX = 10;\nmsg.dY = 10;\nif (msg.startAtoB == null)\n    msg.startAtoB == true;\nif (msg.startAtoC == null)\n    msg.startAtoC == true;\nif (msg.startAtoD == null)\n    msg.startAtoD == true;\nif (msg.startBtoA == null)\n    msg.startBtoA == true;\nif (msg.startBtoC == null)\n    msg.startBtoC == true;\nif (msg.startBtoD == null)\n    msg.startBtoD == true;\nif (msg.startCtoA == null)\n    msg.startCtoA == true;\nif (msg.startCtoB == null)\n    msg.startCtoB == true;\nif (msg.startCtoD == null)\n    msg.startCtoD == true;\nif (msg.startDtoA == null)\n    msg.startDtoA == true;\nif (msg.startDtoB == null)\n    msg.startDtoB == true;\nif (msg.startDtoC == null)\n    msg.startDtoC == true;\n\nif (msg.raspX == null && msg.raspY == null){\n    msg.raspX == 2;\n    msg.raspY == 2;\n}\n\n// Destination Zone D\nif (msg.zoneD == true){\n    \n    // Diagonal transportation section begins\n    // zone A to D transportation begins\n    if (msg.destA == true){\n        if (msg.startAtoD == true)\n            msg.AtoD = 1;\n        do {\n            msg.raspX += 1; //this is where motor will be actuated\n            \n        } while (msg.raspX < 5 && msg.AtoD == 1)\n        if (msg.raspX == 5 && msg.raspY == 2)\n            msg.AtoD = 2;\n        do {\n            //this is where there will be  one-time car rotation\n            msg.raspY += 1; //this is where the motor will be actuted\n        } while (msg.raspY < 8 && msg.AtoD == 2)\n        if (msg.raspY == 8 && msg.raspX == 5)\n            msg.AtoD = 3;\n        do { \n            //this is where there will be  one-time car rotation\n            msg.raspX += 1; //this is where motor will be actuated\n            \n        } while (msg.raspX < 8 && msg.AtoD == 3)    \n        if (msg.raspX == 8 && msg.raspY == 8)\n            msg.AtoD = 4;        \n        if (msg.AtoD == 4){\n            //initiate 180 degree stationary rotation\n        }            \n    }\n    // zone A to D transportation ends\n    // Diagonal transportation section ends\n    \n    \n    // Longitudinal transportation section begins\n    // Zone B to D transportation begins\n    if (msg.destB == true){\n        if (msg.startBtoD == true)\n            msg.BtoD = 1;\n        do {\n            msg.raspX -= 1; //this is where motor will be actuated\n            \n        } while (msg.raspX > 5 && msg.BtoD == 1)\n        if (msg.raspX == 5 && msg.raspY == 2)\n            msg.BtoD = 2;\n        do {\n            //this is where there will be  one-time car rotation\n            msg.raspY += 1; //this is where the motor will be actuted\n        } while (msg.raspY < 8 && msg.BtoD == 2)\n        if (msg.raspY == 8 && msg.raspX == 5)\n            msg.BtoD = 3;\n        do { \n            //this is where there will be  one-time car rotation\n            msg.raspX += 1; //this is where motor will be actuated\n            \n        } while

(msg.raspX < 8 && msg.BtoD == 3) \n if (msg.raspX == 8 && msg.raspY == 8)\n msg.BtoD = 4; \n if (msg.BtoD == 4){\n //initiate 180 degree stationary rotation\n } \n }\n // zone B to D transportation ends\n // Longitudinal transportation section ends \n \n // Lateral transportation section begins\n // Zone C to D transportation begins\n if (msg.destC == true){\n if (msg.startCtoD == true)\n msg.CtoD = 1;\n do { \n msg.raspX += 1; //this is where motor will be actuated\n \n } while (msg.raspX < 8 && msg.CtoD == 1)\n if (msg.raspX == 8 && msg.raspY == 8)\n msg.CtoD = 2;\n if (msg.CtoD == 2){\n //initiate 180 degree stationary rotation\n }\n }\n // zone C to D transportation ends\n // Lateral transportation section ends \n \n}\n\n// Destination Zone C\nif (msg.zoneD == true){\n \n // Diagonal transportation section begins\n // zone A to D transportation begins\n if (msg.destA == true){\n if (msg.startAtoD == true)\n msg.AtoD = 1;\n do { \n msg.raspX += 1; //this is where motor will be actuated\n \n } while (msg.raspX < 5 && msg.AtoD == 1)\n if (msg.raspX == 5 && msg.raspY == 2)\n msg.AtoD = 2;\n do {\n //this is where there will be one-time car rotation\n msg.raspY += 1; //this is where the motor will be actuted\n } while (msg.raspY < 8 && msg.AtoD == 2)\n if (msg.raspY == 8 && msg.raspX == 5)\n msg.AtoD = 3;\n do { \n //this is where there will be one-time car rotation\n msg.raspX += 1; //this is where motor will be actuated\n \n } while (msg.raspX < 8 && msg.AtoD == 3) \n if (msg.raspX == 8 && msg.raspY == 8)\n msg.AtoD = 4; \n if (msg.AtoD == 4){\n //initiate 180 degree stationary rotation\n } \n }\n // zone A to D transportation ends\n // Diagonal transportation section ends\n \n // Longitudinal transportation section begins\n // Zone B to D transportation begins\n if (msg.destB == true){\n if (msg.startBtoD == true)\n msg.BtoD = 1;\n do { \n msg.raspX -= 1; //this is where motor will be actuated\n \n } while (msg.raspX > 5 && msg.BtoD == 1)\n if (msg.raspX == 5 && msg.raspY == 2)\n msg.BtoD = 2;\n do {\n //this is where there will be one-time car rotation\n msg.raspY += 1; //this is where the motor will be actuted\n } while (msg.raspY < 8 && msg.BtoD == 2)\n if (msg.raspY == 8 && msg.raspX == 5)\n msg.BtoD = 3;\n do { \n //this is where there will be one-time car rotation\n msg.raspX += 1; //this is where motor will be actuated\n \n } while (msg.raspX < 8 && msg.BtoD == 3) \n if (msg.raspX == 8 && msg.raspY == 8)\n msg.BtoD = 4; \n if (msg.BtoD == 4){\n //initiate 180 degree stationary rotation\n } \n }\n // zone B to D transportation ends\n // Longitudinal transportation section ends \n \n // Lateral transportation section begins\n // Zone C to D transportation begins\n if (msg.destC == true){\n if (msg.startCtoD == true)\n msg.CtoD = 1;\n do { \n msg.raspX += 1; //this is where motor will be actuated\n \n } while (msg.raspX < 8 && msg.CtoD == 1)\n if (msg.raspX == 8 && msg.raspY == 8)\n msg.CtoD = 2;\n if (msg.CtoD == 2){\n //initiate 180 degree stationary rotation\n }\n }\n // zone C to D transportation ends\n // Lateral transportation section ends \n \n}\n\n// Destination Zone B\nif (msg.zoneD == true){\n \n // Diagonal transportation section begins\n // zone A to D transportation begins\n if (msg.destA == true){\n if (msg.startAtoD == true)\n msg.AtoD = 1;\n do { \n msg.raspX += 1; //this is where motor will be actuated\n \n } while (msg.raspX < 5 && msg.AtoD == 1)\n if (msg.raspX == 5 && msg.raspY == 2)\n msg.AtoD = 2;\n do {\n //this is where there will be one-time car rotation\n msg.raspY += 1; //this is where the motor will be actuted\n } while (msg.raspY < 8 && msg.AtoD == 2)\n if (msg.raspY == 8 && msg.raspX == 5)\n msg.AtoD = 3;\n do { \n //this is where there will be one-time car rotation\n msg.raspX += 1; //this is where motor will be actuated\n \n } while (msg.raspX < 8 && msg.AtoD == 3) \n if (msg.raspX == 8 && msg.raspY == 8)\n msg.AtoD = 4; \n if (msg.AtoD == 4){\n //initiate 180 degree stationary rotation\n } \n }\n // zone A to D transportation ends\n // Diagonal transportation section ends\n \n \n // Longitudinal transportation section begins\n // Zone B to D transportation begins\n if (msg.destB == true){\n if (msg.startBtoD == true)\n msg.BtoD = 1;\n do { \n msg.raspX -= 1; //this is where motor will be actuated\n \n } while (msg.raspX > 5 && msg.BtoD == 1)\n if (msg.raspX == 5 && msg.raspY == 2)\n msg.BtoD = 2;\n do {\n //this is where there will be one-time car rotation\n msg.raspY += 1; //this is where the motor will be actuted\n } while (msg.raspY < 8 && msg.BtoD == 2)\n if (msg.raspY == 8 && msg.raspX == 5)\n msg.BtoD = 3;\n do { \n //this is where there will be one-time car rotation\n msg.raspX += 1; //this is where motor will be actuated\n \n } while (msg.raspX < 8 && msg.BtoD == 3) \n if (msg.raspX == 8 && msg.raspY == 8)\n msg.BtoD = 4; \n if (msg.BtoD == 4){\n //initiate 180 degree stationary rotation\n } \n }\n // zone B to D transportation ends\n // Longitudinal transportation section ends \n \n // Lateral transportation section begins\n // Zone C to D transportation begins\n if (msg.destC == true){\n if (msg.startCtoD == true)\n msg.CtoD = 1;\n do { \n msg.raspX += 1; //this is where motor will be actuated\n \n } while (msg.raspX < 8 && msg.CtoD == 1)\n if (msg.raspX == 8 && msg.raspY == 8)\n msg.CtoD = 2;\n if (msg.CtoD == 2){\n //initiate 180 degree stationary rotation\n }\n }\n // zone C to D transportation ends\n // Lateral transportation section ends \n \n}\n\n// Destintion Zone A\nif (msg.zoneD == true){\n \n // Diagonal transportation section begins\n // zone A to D transportation begins\n if (msg.destA == true){\n if (msg.startAtoD == true)\n msg.AtoD = 1;\n do { \n msg.raspX += 1; //this is where motor will be actuated\n \n } while (msg.raspX < 5 && msg.AtoD == 1)\n if (msg.raspX == 5 && msg.raspY == 2)\n msg.AtoD = 2;\n do {\n //this is where there will be one-time car rotation\n msg.raspY += 1; //this is where the motor will be actuted\n } while (msg.raspY < 8 && msg.AtoD == 2)\n if (msg.raspY == 8 && msg.raspX == 5)\n msg.AtoD = 3;\n do { \n //this is where there will be one-time car rotation\n msg.raspX += 1; //this is where motor will be actuated\n \n } while (msg.raspX < 8 && msg.AtoD == 3) \n if (msg.raspX == 8 && msg.raspY == 8)\n msg.AtoD = 4; \n if (msg.AtoD == 4){\n //initiate 180 degree stationary rotation\n } \n }\n // zone A to D transportation ends\n // Diagonal transportation section ends\n \n \n // Longitudinal transportation section begins\n // Zone B to D transportation begins\n if (msg.destB == true){\n if (msg.startBtoD == true)\n msg.BtoD = 1;\n do { \n msg.raspX -= 1; //this is where motor will be actuated\n \n } while (msg.raspX > 5 && msg.BtoD == 1)\n if (msg.raspX == 5 && msg.raspY == 2)\n msg.BtoD = 2;\n do {\n //this is where there will be one-time car rotation\n msg.raspY += 1; //this is where the motor will be actuted\n } while (msg.raspY < 8 && msg.BtoD == 2)\n if (msg.raspY == 8 && msg.raspX == 5)\n msg.BtoD = 3;\n do { \n //this is where there will be one-time car rotation\n msg.raspX += 1; //this is where motor will be actuated\n \n } while (msg.raspX < 8 && msg.BtoD == 3) \n if (msg.raspX == 8 && msg.raspY == 8)\n msg.BtoD = 4; \n if (msg.BtoD == 4){\n //initiate 180 degree stationary rotation\n } \n }\n // zone B to D transportation ends\n // Longitudinal transportation section ends \n \n // Lateral transportation section begins\n // Zone C to D transportation begins\n if (msg.destC == true){\n if (msg.startCtoD == true)\n msg.CtoD = 1;\n do { \n msg.raspX += 1; //this is where motor will be

actuated\n \n } while (msg.raspX < 8 && msg.CtoD == 1)\n if (msg.raspX == 8 && msg.raspY == 8)\n msg.CtoD = 2;\n if (msg.CtoD == 2){\n //initiate 180 degree stationary rotation\n }\n }\n // zone C to D transportation ends\n // Lateral transportation section ends \n \n}\nreturn msg;","outputs":1,"noerr":0,"initialize":"","finalize":"","x":750,"y":582.6667060852051,"wires":[[]]},{"id":"eb0412ef.fbbfc","type":"function","z":"c994d44a.0d89e8","name":"","func":"var sLength = msg.payload.length;\n\nfor (var i = 0; i<sLength; i++){\n\n if (msg.payload[i] == 'A' && msg.payload[i+1] == 'd' && msg.payload[i+2] == 'd' && msg.payload[i+3] == 'r' && msg.payload[i+4] == 'e' && msg.payload[i+5] == 's' && msg.payload[i+6] == 's'){\n \n msg.add2 =msg.payload[i+9] + msg.payload[i+10] + msg.payload[i+11] + msg.payload[i+12] + msg.payload[i+13] + msg.payload[i+14] + msg.payload[i+15] + msg.payload[i+16] + msg.payload[i+17] + msg.payload[i+18] + msg.payload[i+19] + msg.payload[i+20] + msg.payload[i+21] + msg.payload[i+22] + msg.payload[i+23] + msg.payload[i+24] + msg.payload[i+25];\n ///for (var = k; k<msg.add2.length; k++){\n if (msg.add2[0] == '9' && msg.add2[1] == '8'){// && msg.add2[2] == \":\" && msg.add2[3] == '2' && msg.add2[4] == 'c' && msg.add2[5] == \":\" && msg.add2[6] == 'b' && msg.add2[7] == '6' && msg.add2[8] == \":\" && msg.add2[9] == '4' && msg.add2[10] == '9' && msg.add2[11] == \":\" && msg.add2[12] == 'e' && msg.add2[13] == '1' && msg.add2[14] == \":\" && msg.add2[15] == '4' && msg.add2[16] == 'f' ){\n msg.add3 = msg.add2;\n msg.test = \"aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa\";\n msg.sig = msg.payload[109] + msg.payload[110];//116,117\n \n }\n \n \n //}\n /*if (msg.add2 == \"6e:91:44:e1:e1:a9\"){//\"08:2C:B6:49:E1:4F\"){\n msg.add = msg.add2;\n msg.sig = \"61\";\n for (var j = 0; j<sLength; j++){\n if (msg.payload[j] == 'R' && msg.payload[j+1] == 's' && msg.payload[j+2] == 's' && msg.payload[j+3] == 'i'){\n msg.sig = \"52\";//msg.payload[j+7] + msg.payload[j+8];\n \n //msg.payload = \"aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa\";\n }\n }\n }*/\n }\n }\n \n \n\nreturn msg;","outputs":1,"noerr":0,"initialize":"","finalize":"","x":557.0000152587891,"y":108.6666669845581,"wires":[[]]},{"id":"45ab6da9.a6cca4","type":"function","z":"c994d44a.0d89e8","name":"","func":"var sLength = msg.payload.length;\n\nfor (var i = 0; i<sLength; i++){\n\n if (msg.payload[i] == 'A' && msg.payload[i+1] == 'd' && msg.payload[i+2] == 'd' && msg.payload[i+3] == 'r' && msg.payload[i+4] == 'e' && msg.payload[i+5] == 's' && msg.payload[i+6] == 's'){\n \n msg.add2 =msg.payload[i+9] + msg.payload[i+10] + msg.payload[i+11] + msg.payload[i+12] + msg.payload[i+13] + msg.payload[i+14] + msg.payload[i+15] + msg.payload[i+16] + msg.payload[i+17] + msg.payload[i+18] + msg.payload[i+19] + msg.payload[i+20] + msg.payload[i+21] + msg.payload[i+22] + msg.payload[i+23] + msg.payload[i+24] + msg.payload[i+25];\n ///for (var = k; k<msg.add2.length; k++){\n if (msg.add2[0] == 'D' && msg.add2[1] == '8'){// && msg.add2[2] == \":\" && msg.add2[3] == '2' && msg.add2[4] == 'c' && msg.add2[5] == \":\" && msg.add2[6] == 'b' && msg.add2[7] == '6' && msg.add2[8] == \":\" && msg.add2[9] == '4' && msg.add2[10] == '9' && msg.add2[11] == \":\" && msg.add2[12] == 'e' && msg.add2[13] == '1' && msg.add2[14] == \":\" && msg.add2[15] == '4' && msg.add2[16] == 'f' ){\n msg.add3 = msg.add2;\n msg.test = \"aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa\";\n msg.sig = msg.payload[109] + msg.payload[110];//116,117\n \n }\n \n \n //}\n /*if (msg.add2 ==

\"6e:91:44:e1:e1:a9\"){//\"08:2C:B6:49:E1:4F\"){\n msg.add = msg.add2;\n msg.sig = \"61\";\n for (var j = 0; j<sLength; j++){\n if (msg.payload[j] == 'R' && msg.payload[j+1] == 's' && msg.payload[j+2] == 's' && msg.payload[j+3] == 'i'){\n msg.sig = \"52\";//msg.payload[j+7] + msg.payload[j+8];\n \n //msg.payload = \"aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa\";\n }\n }\n }*/\n }\n }\n \n \n\nreturn msg;","outputs":1,"noerr":0,"initialize":"","finalize":"","x":670.3333129882812,"y":208.6666717529297,"wires":[[]]},{"id":"62e812dc.b3f72c","type":"debug","z":"c994d44a.0d89e8","name":"","active":true,"tosidebar":true,"console":false,"tostatus":false,"complete":"sig","targetType":"msg","statusVal":"","statusType":"auto","x":923.0000038146973,"y":144.66667461395264,"wires":[]},{"id":"910f87cf.9376b8","type":"ui_group","name":"Default","tab":"76256355.0eab0c","disp":true,"width":"6"},{"id":"93485f7d.0073f","type":"serial-port","serialport":"/dev/ttyUSB0","serialbaud":"115200","databits":"8","parity":"none","stopbits":"1","waitfor":"","newline":"\\n\\r","bin":"false","out":"char","addchar":"false","responsetimeout":""},{"id":"76256355.0eab0c","type":"ui_tab","name":"Signal Meter","icon":"dashboard"}]

# Appendix C – Project Photos

This appendix contains physical images of the project.

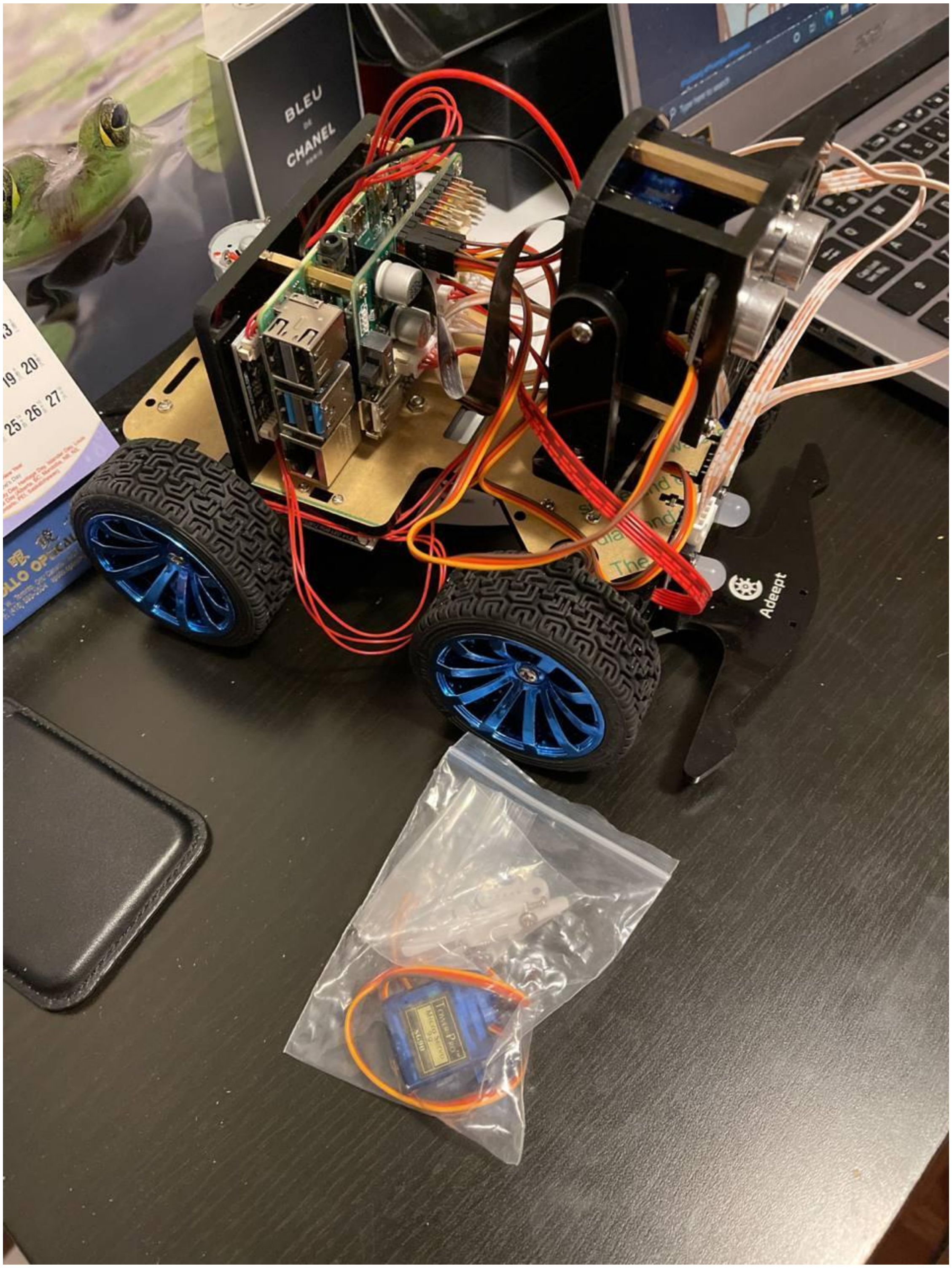



*Figure 20 Top-view of the Raspberry Pi Car.*

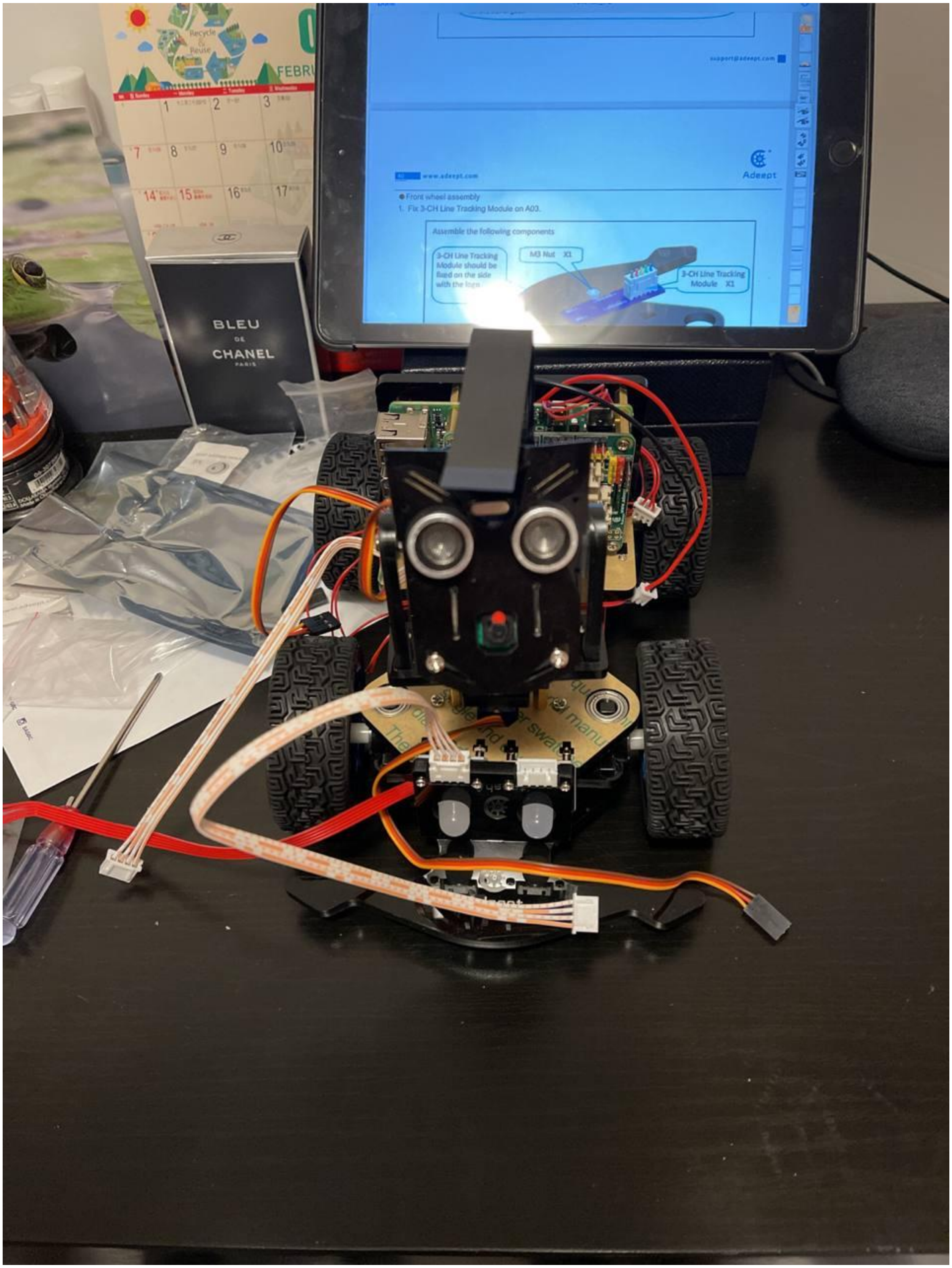



*Figure 21 Front-view of the Raspberry Pi Car.*

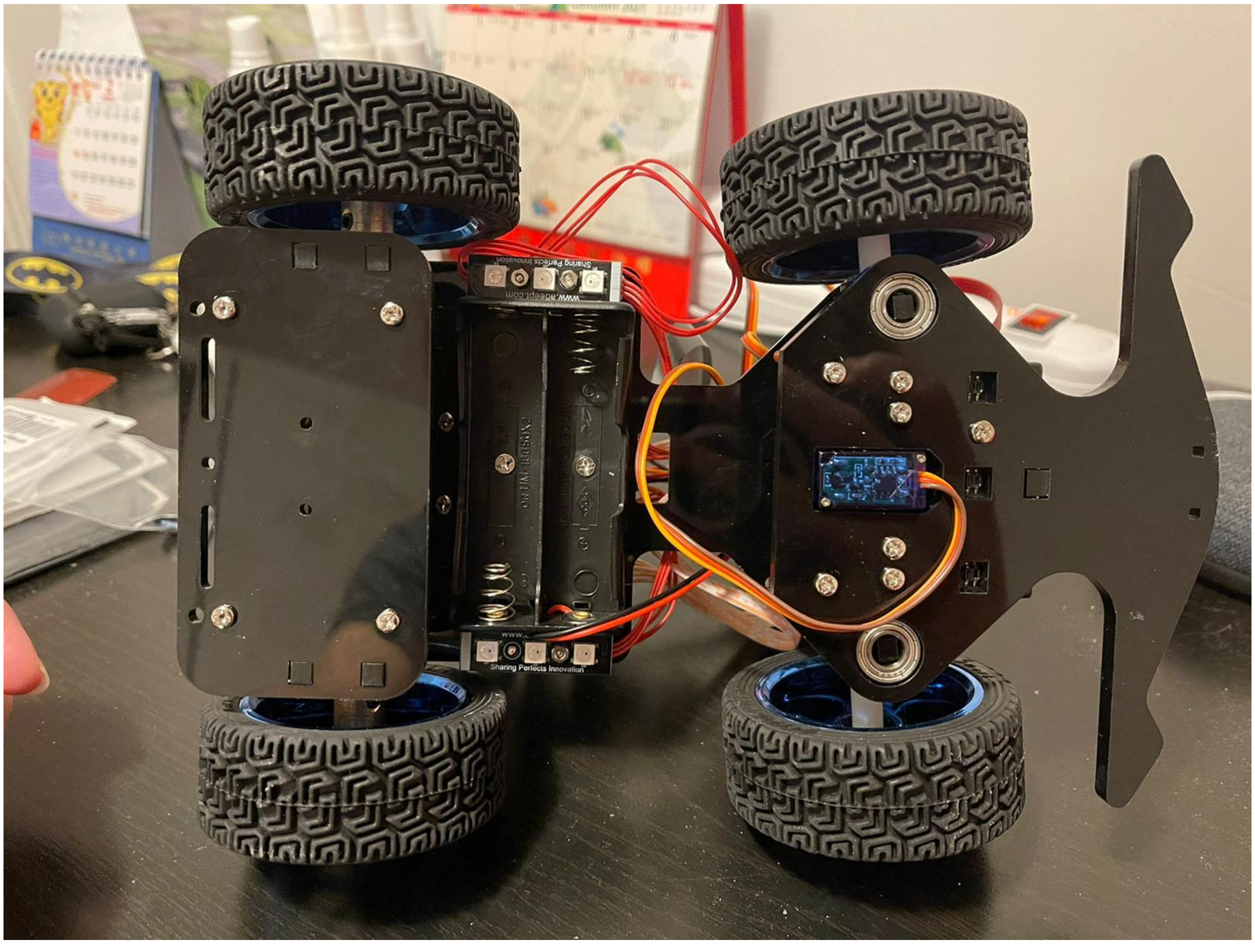



*Figure 22 Bottom-view of the Raspberry Pi Car.*

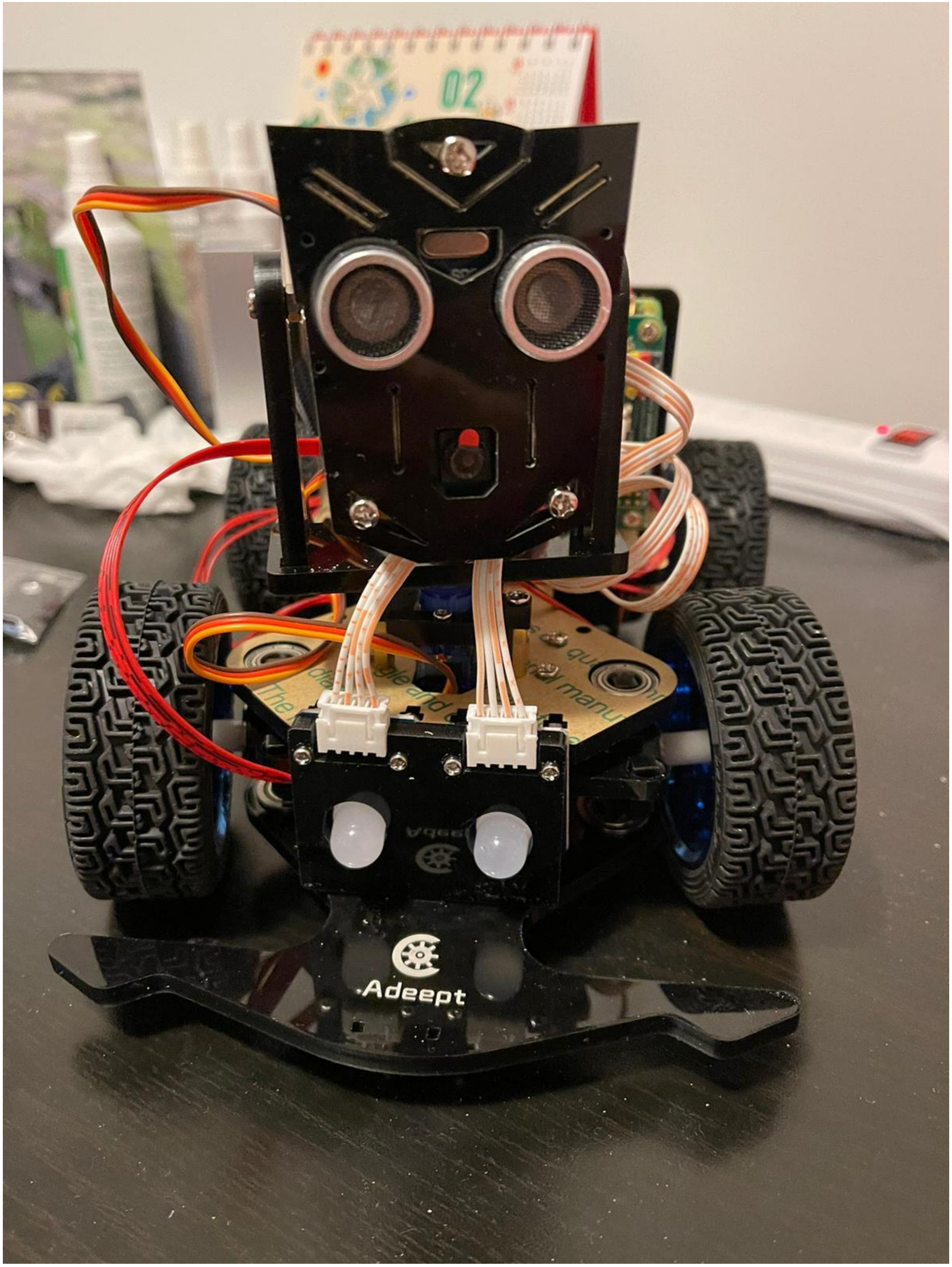



*Figure 23 Another front-view of the Raspberry Pi Car.*

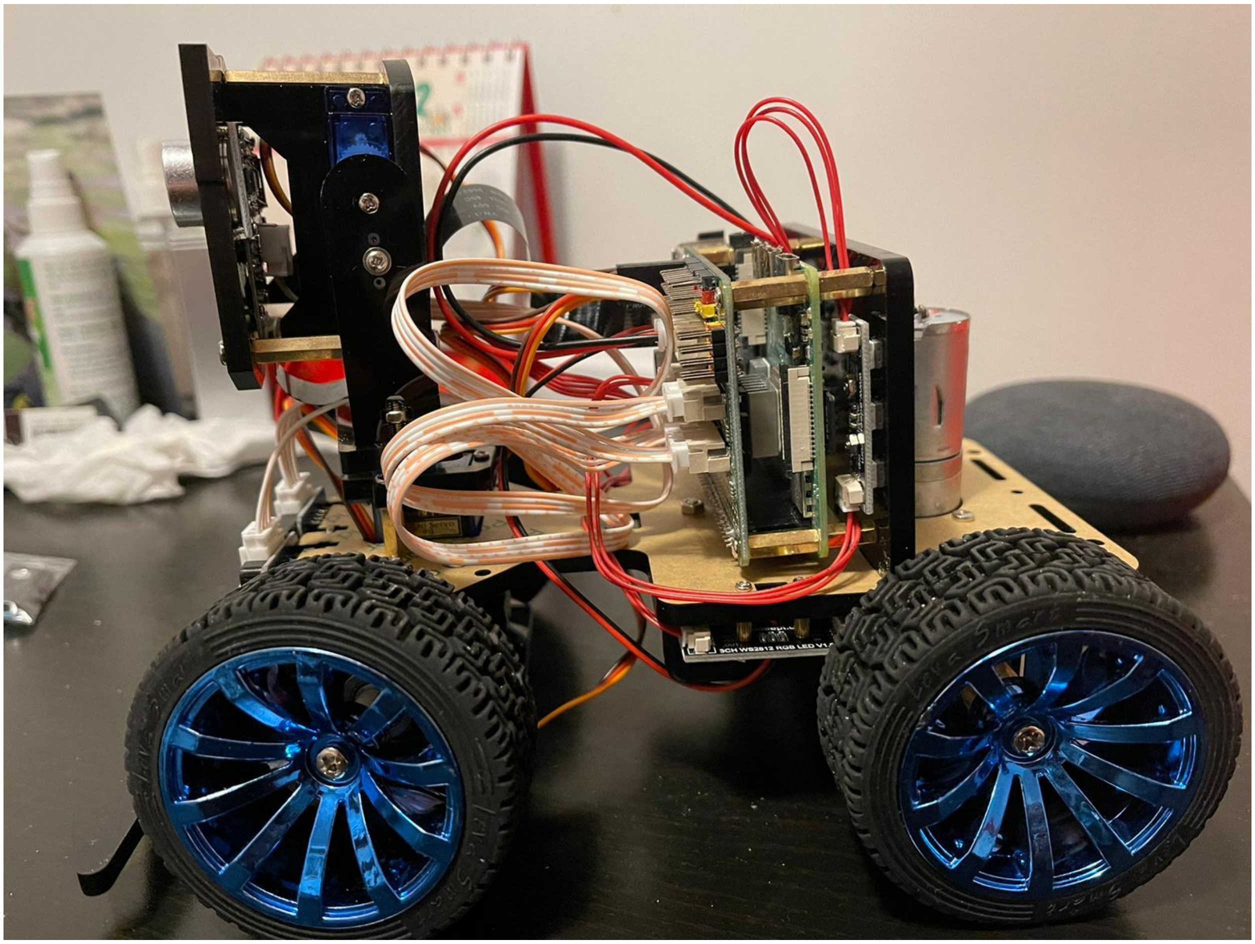


*Figure 24 Side-view of the Raspberry Pi Car.*

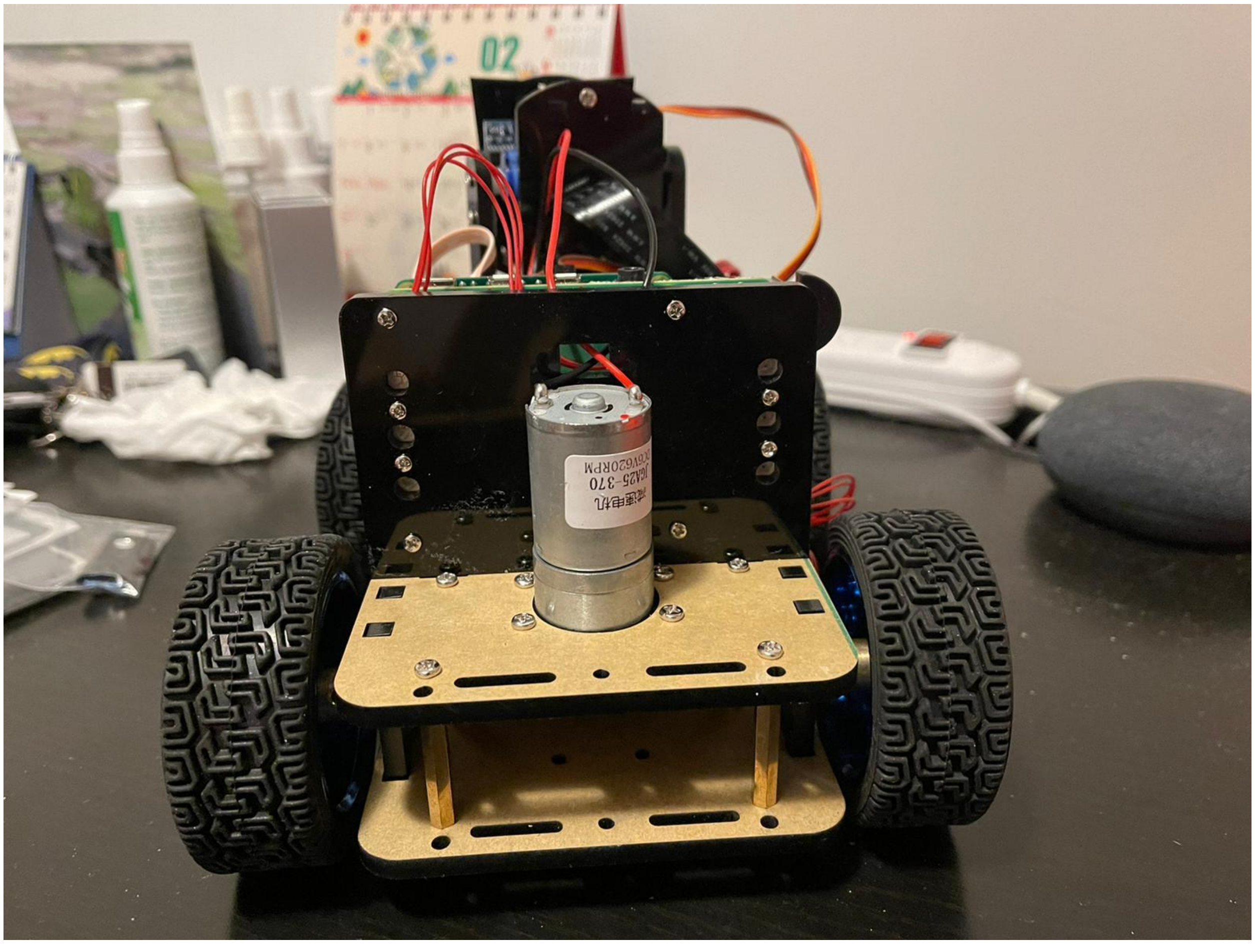


*Figure 25 Back-view of the Raspberry Pi Car.*

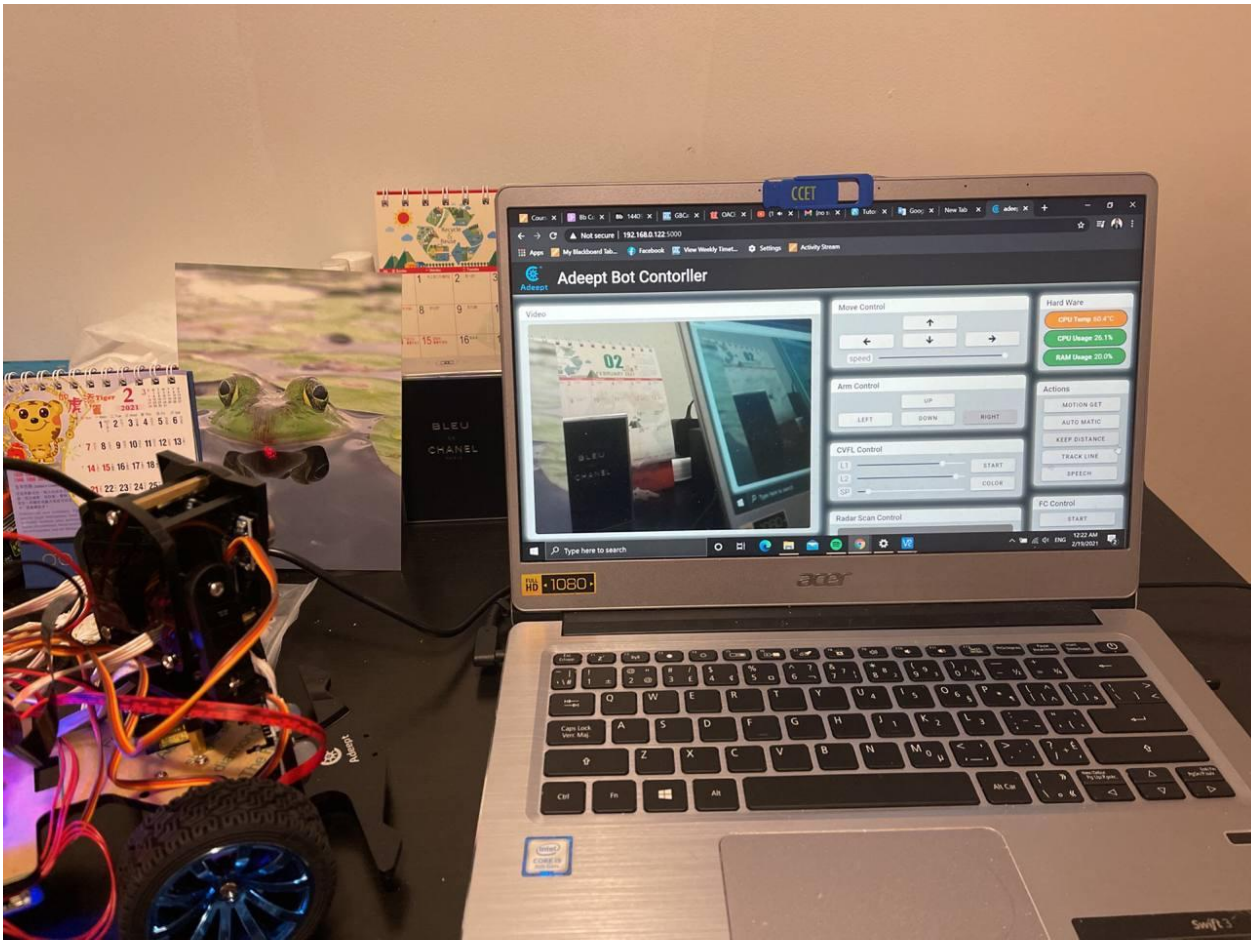



*Figure 26 Camera and control panel for the Raspberry Pi Car.*

# Appendix D - Schematics

This appendix contains the project schematics.

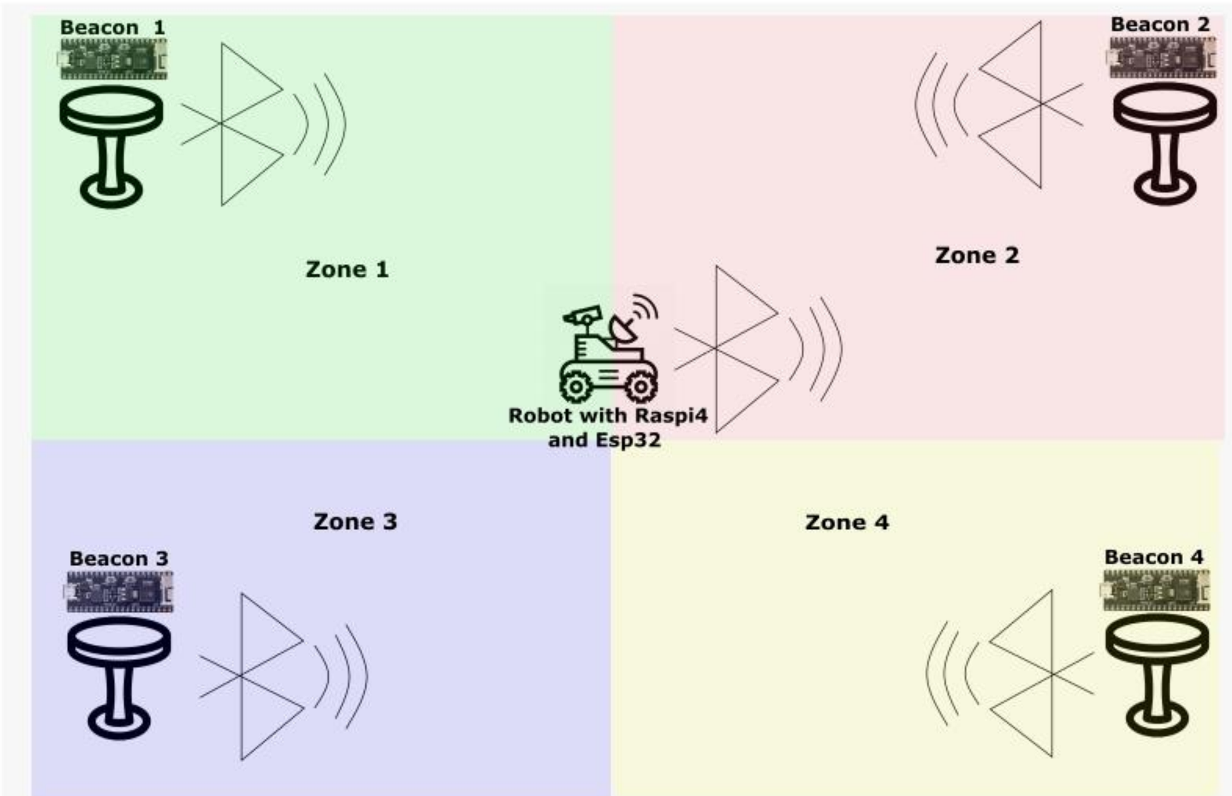



*Figure 27 Project schematic with four ESP32 Bluetooth beacons, the Raspberry Pi car and zone indications.*

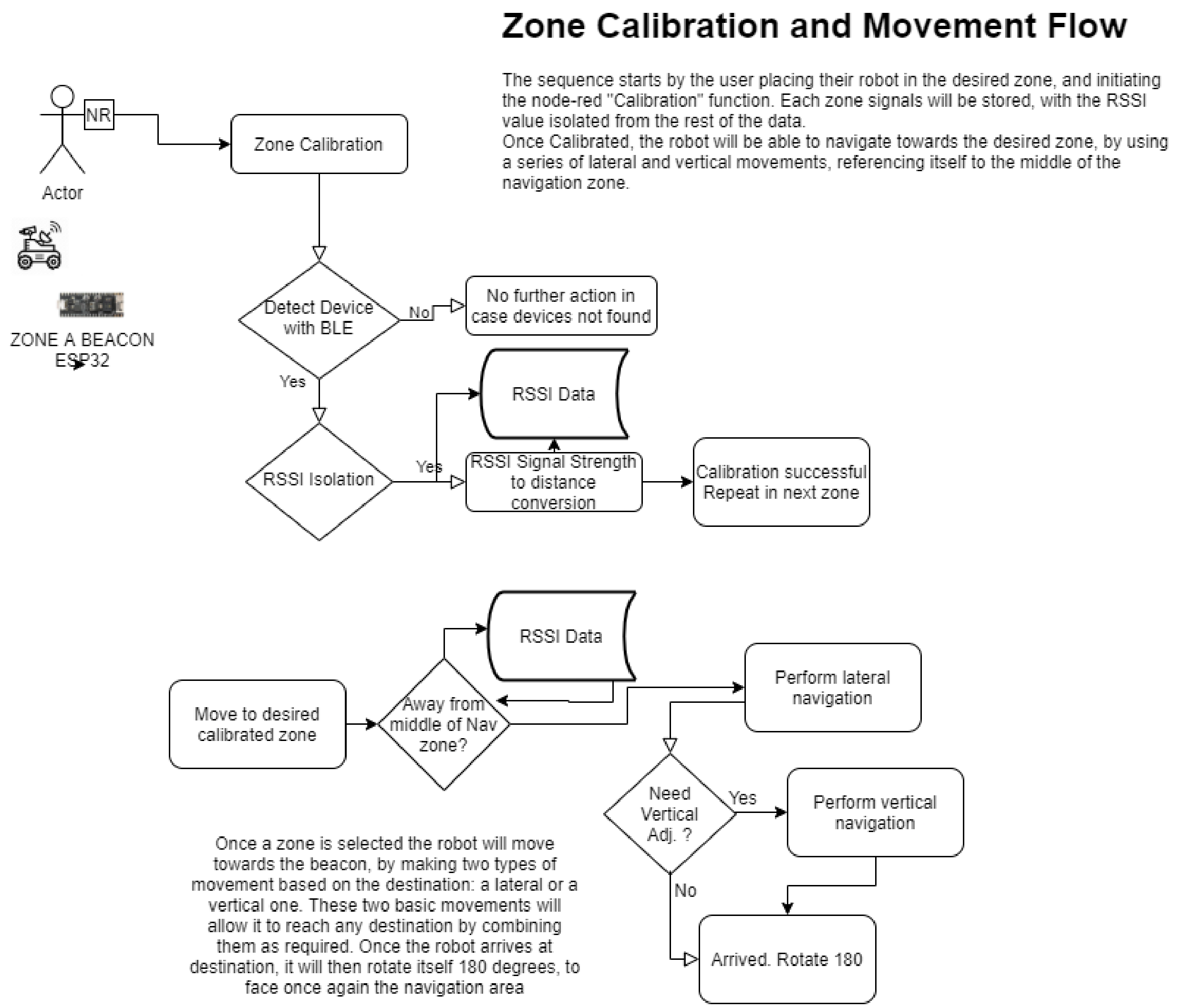



*Figure 28 Flow chart indicating the logical process of movement and calibrating zones*

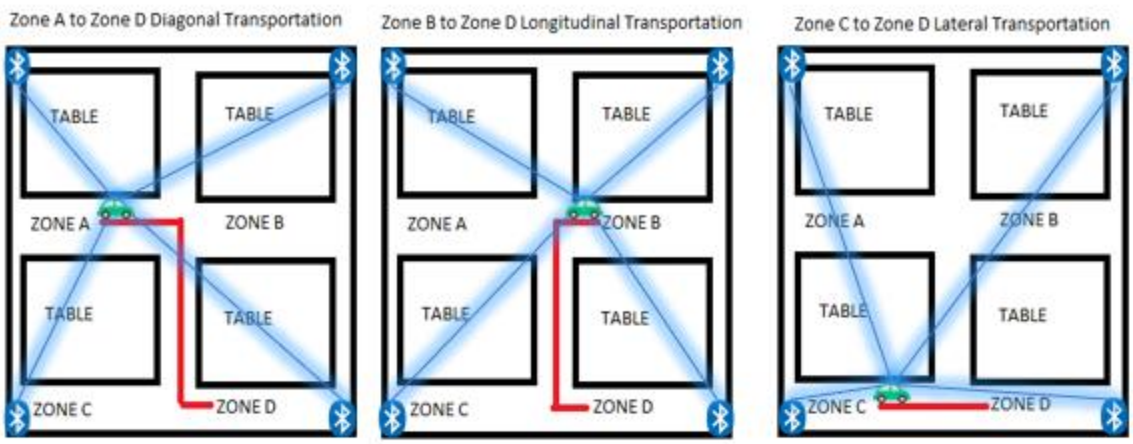



*Figure 29 Diagram displaying the three types of movement in order to travel to Zone D: Lateral, Longitudinal, and Diagonal*

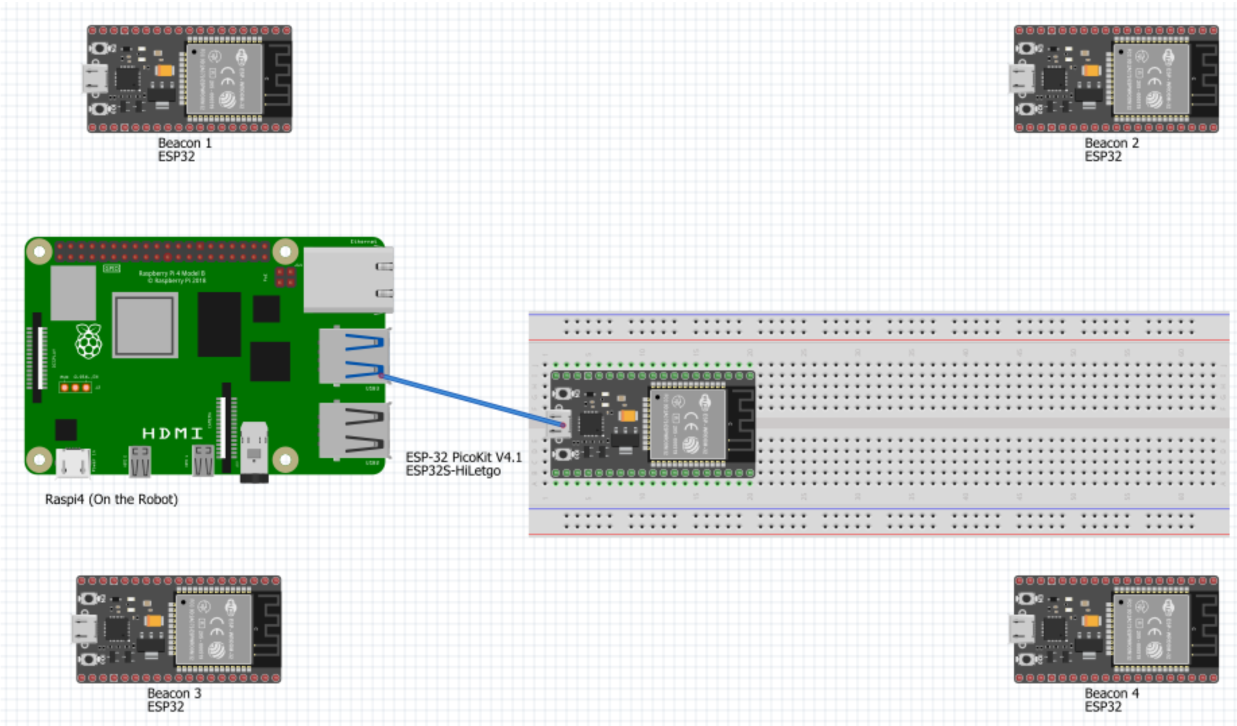



*Figure 30Schematic made with Fritzing that shows the components involved in the project. We will be using the robot's onboard raspi4 along an esp32 to scan, detect, and infer its distance from the other Esp32s that function as beacons.*

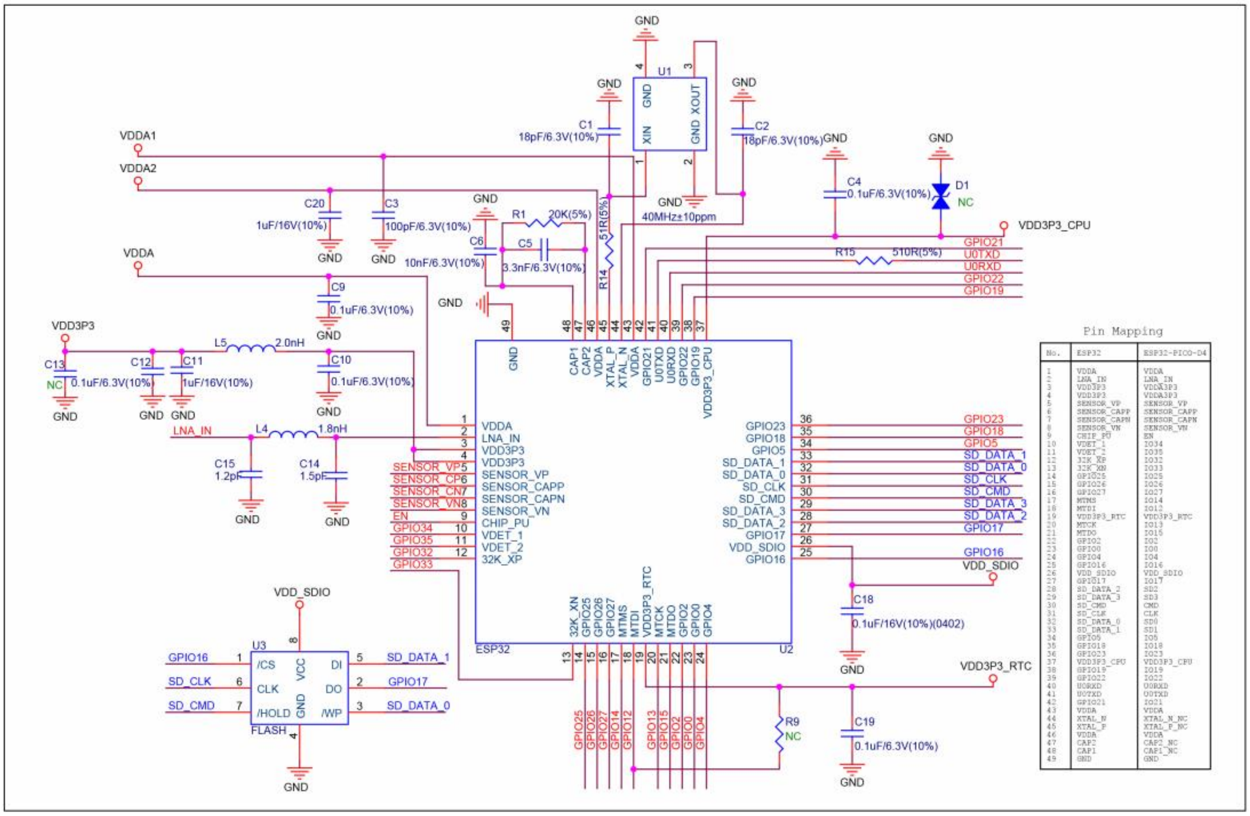
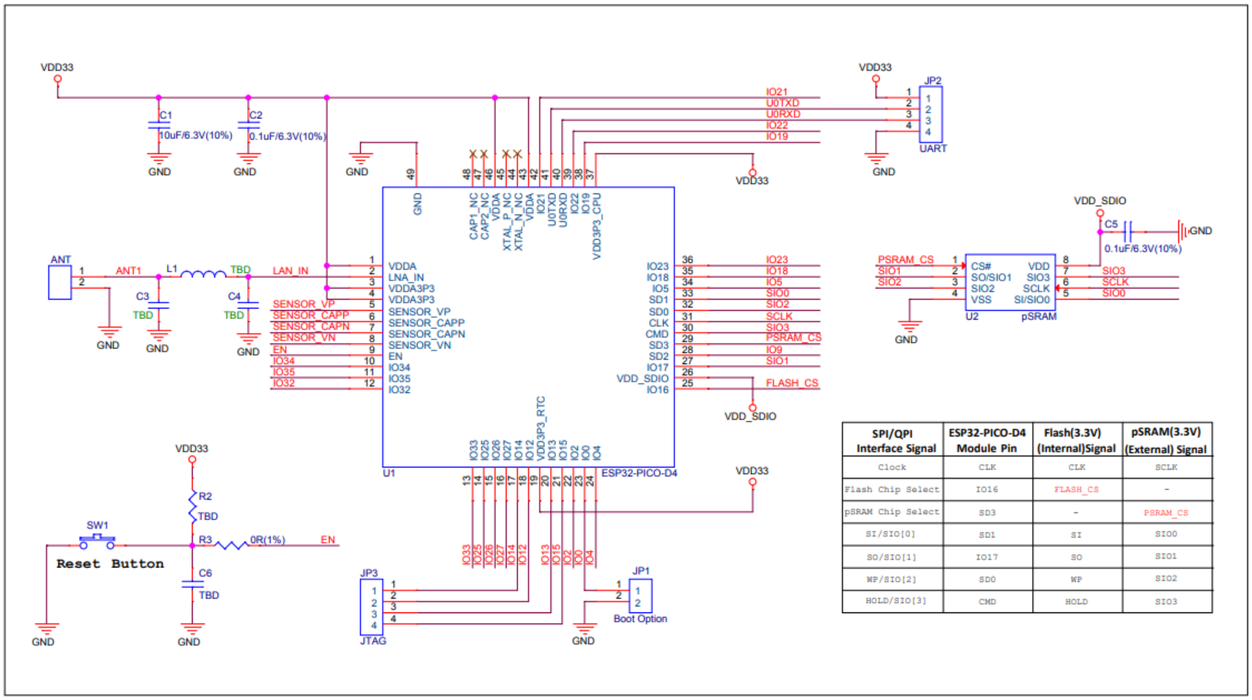



*Figure 31 ESP32-PICO-D4 Module Schematics.*




| SPI/QPI Interface Signal | ESP32-PICO-D4 Module Pin | Flash(3.3V) (Internal)Signal | pSRAM(3.3V) (External) Signal |
|---|---|---|---|
| Clock | CLK | CLK | SCLK |
| Flash Chip Select | IO16 | FLASH_CS | - |
| pSRAM Chip Select | SD3 | - | PSRAM_CS |
| SI/SIO[0] | SD1 | SI | SIO0 |
| SO/SIO[1] | IO17 | SO | SIO1 |
| WP/SIO[2] | SD0 | WP | SIO2 |
| HOLD/SIO[3] | CMD | HOLD | SIO3 |

*Figure 32 ESP32-PICO-D4 Module Peripheral Schematics.*

# Appendix E - BoM

This appendix contains the bill of materials for the project.

| Product | Price | Quantity |
|---|---|---|
| 24 Pack AAA Batteries | 15.15 | 1 |
| Micro USB to USB | 3.73 | 1 |
| Battery Case | 11.8 | (price on left includes 4) |
| Directional Antenna w/ SMA | 21.8 | (price on left includes 4) |
| Omnidirectional Antenna w/ SMA | 7.96 | 1 |
| ESP32 | 63.65 | (price on left includes 5) |
| Raspi 3 (1GB) | 72.95 | 1 |
| microSD card | 18.5 | 1 |
| Tablet | Free from School | 1 |
| **Total before tax** | **215.54** | |
| **Total after tax** | **243.5602** | |

# Appendix F – GANTT & Project Poster

This appendix contains the project GANTT chart.

**PROJECT TITLE**

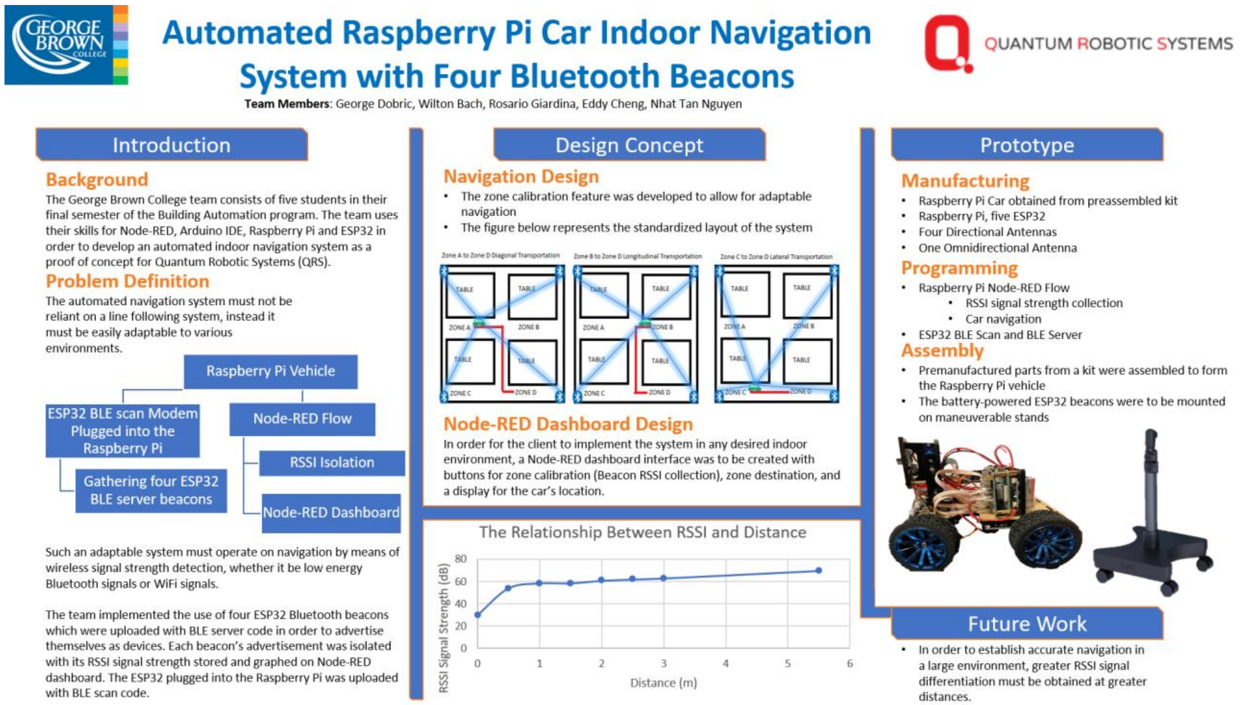
Company Name
Project Lead

SIMPLE GANTT CHART by Vertex42.com
https://www.vertex42.com/ExcelTemplates/simple-gantt-chart.html

Project Start: Fri, 1-15-2021

Display Week: 8

| TASK | START | END |
|---|---|---|
| **Phase1 (Planning)** | | |
| Brainstorming Project Ideas | 1-15-21 | 1-15-21 |
| Choosing Project | 1-15-21 | 1-15-21 |
| Research about the Project | 1-15-21 | 1-22-21 |
| Craeting BOM (Bill of Materials) | 1-22-21 | 1-29-21 |
| **Phase 2 (Programming/Preparing)** | | |
| Programming of ESP32 for BLE Communication (Using Arduino) | 1-22-21 | 2-12-21 |
| Assembly of Car Robot Kit | 2-12-21 | 2-19-21 |
| Schematics Arrangement | 2-15-21 | 2-19-21 |
| Programming of ESP32 for BLE Communication (Node-Red) | 2-12-21 | 2-26-21 |
| Midterm Report | 2-26-21 | 2-26-21 |
| Converting RSSI to Distance | 2-26-21 | 3-19-21 |
| Development of Navigation System with 4 Bluetooth Beacons | 3-5-21 | 3-26-21 |
| Calibration of zones | 3-19-21 | 4-2-21 |
| Design of Dashboard/ Display (HMI) | 3-26-21 | 4-2-21 |
| **Phase 3 (Assembly & Testing)** | | |
| Assembly of All Parts in the Environment of Testing | 4-2-21 | 4-9-21 |
| Testing of the Final Assembly | 4-9-21 | 4-16-21 |
| Fixing and Polishing Details & Minor Deffects | 4-9-21 | 4-16-21 |
| Final Testing | 4-16-21 | 4-23-21 |
| Phase 4 (Final Report) | | |
| Final Report | 4-23-21 | 4-23-21 |
| | | |
| *Insert new rows ABOVE this one* | | |


Automated Raspberry Pi Car Indoor Navigation System with Four Bluetooth Beacons
Team Members: George Dobric, Wilton Bach, Rosario Giardina, Eddy Cheng, Nhat Tan Nguyen
QUANTUM ROBOTIC SYSTEMS
Introduction
Background
The George Brown College team consists of five students in their final semester of the Building Automation program. The team uses their skills for Node-RED, Arduino IDE, Raspberry Pi and ESP32 in order to develop an automated indoor navigation system as a proof of concept for Quantum Robotic Systems (QRS).
Problem Definition
The automated navigation system must not be reliant on a line following system, instead it must be easily adaptable to various environments.
Raspberry Pi Vehicle
ESP32 BLE scan Modem Plugged into the Raspberry Pi
Gathering four ESP32 BLE server beacons
Node-RED Flow
RSSI Isolation
Node-RED Dashboard
Such an adaptable system must operate on navigation by means of wireless signal strength detection, whether it be low energy Bluetooth signals or WiFi signals.
The team implemented the use of four ESP32 Bluetooth beacons which were uploaded with BLE server code in order to advertise themselves as devices. Each beacon's advertisement was isolated with its RSSI signal strength stored and graphed on Node-RED dashboard. The ESP32 plugged into the Raspberry Pi was uploaded with BLE scan code.
Design Concept
Navigation Design
• The zone calibration feature was developed to allow for adaptable navigation
• The figure below represents the standardized layout of the system
Zone A to Zone D Diagonal Transportation
Zone B to Zone D Longitudinal Transportation
Zone C to Zone D Lateral Transportation
TABLE
ZONE A
ZONE B
ZONE C
ZONE D
Node-RED Dashboard Design
In order for the client to implement the system in any desired indoor environment, a Node-RED dashboard interface was to be created with buttons for zone calibration (Beacon RSSI collection), zone destination, and a display for the car's location.
The Relationship Between RSSI and Distance
RSSI Signal Strength (dB)
0
20
40
60
80
Distance (m)
0
1
2
3
4
5
6
Prototype
Manufacturing
• Raspberry Pi Car obtained from preassembled kit
• Raspberry Pi, five ESP32
• Four Directional Antennas
• One Omnidirectional Antenna
Programming
• Raspberry Pi Node-RED Flow
• RSSI signal strength collection
• Car navigation
• ESP32 BLE Scan and BLE Server
Assembly
• Premanufactured parts from a kit were assembled to form the Raspberry Pi vehicle
• The battery-powered ESP32 beacons were to be mounted on maneuverable stands
Future Work
• In order to establish accurate navigation in a large environment, greater RSSI signal differentiation must be obtained at greater distances.